NASA-CR-194197

INTELLIGENT GAS TUNGSTEN ARC WELDING CONTROL

- SBIR PHASE II FINAL REPORT -

NASA Contract No. NAS8-37401

Release Date: 4-27-92
Proposal #:
SBIR-04.08-0960

Mid-South Engineering, Inc.
2131 Belcourt Avenue
Nashville, Tennessee 37212

SUMMARY

The primary objective of this research was to design a control architecture for control of the Gas Tungsten Arc Welding (GTAW) process, and implement it in a prototype welding controller, to be delivered to NASA at the end of the research period. Specifically, some common problems, or potential problems, of the GTAW process were discussed with NASA and Rocketdyne personnel at the outset of the work, and the focus of the controller design was set on these specific items. The key issues addressed in the design of the delivered controller were: (i) enhanced control of the GTAW equipment parameters, and (ii) enhanced control of the Automatic Voltage Control (AVC) subsystem. These objectives were selected and approached with the welding tasks on the Space Shuttle Main Engine (SSME) in mind. Furthermore, all work and equipment selection was tailored to be adaptable to Variable Polarity Plasma Arc Welding (VPPAW) control at a later time, if desired.

The research consisted of (i) review and study of existing approaches to enhanced arc welding control, (ii) research and conduction of experiments to examine the feasibility of possible solutions, and (iii) design and implementation of the selected techniques.

Extensive research of existing methods for arc welding modeling and control was carried out, and some of the key findings are discussed in this report. Numerous simulations and experiments were carried out to determine which techniques were viable and which should be rejected in the final implementation.

Selection of equipment and software items turned out to be less trivial than anticipated. A computer architecture based on the VME-bus, but using the Intel 80286 and 80386 processors, was finally selected. This approach combines the robustness of the industrial VME-bus and the software sophistication, economics, and availability of the MS-DOS/WINDOWS environment.

The main objective of the work, the design of a control system and its implementation in a process control computer, has been accomplished.

(NASA-CR-194197) INTELLIGENT GAS TUNGSTEN ARC WELDING CONTROL Final Report (Mid-South Engineering) 162 p

N94-70559

Unclas

Z9/37 0183171

INTELLIGENT GAS TUNGSTEN ARC WELDING CONTROL

- SBIR PHASE II FINAL REPORT -

Produced by

Mid-South Engineering, Inc.

2131 Belcourt Avenue

Nashville, Tennessee 37212

Mid-South Engineering, Inc: Kristinn Andersen
Robert Joel Barnett
James F. Springfield

Vanderbilt University - Subcontract: Dr. George E. Cook
Dr. Alvin M. Strauss
Jon B. Bjorgvinsson

Contract No. NAS8-37401

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

ARMA	- Auto-Regressive-Moving-Average
AVC	- Automatic Voltage Control
C	- Capacitance
cfh	- Cubic Feet per Hour
DWP	- Direct Weld Parameters
DWP_{REF}	- Requested or reference DWP values
E	- Voltage
GMAW	- Gas Metal Arc Welding
GTAW	- Gas Tungsten Arc Welding
ipm	- Inches per minute
I	- Electrical current
IWP	- Indirect Weld Parameters
K	- Amplifier gain
K_v	- Arc voltage sensitivity to arc length
L_{arc}	- Arc length
Q_{in}	- Input heat flow
T	- Temperature variable
TWD	- Tip-To-Workpiece Distance
v	- Speed (e.g. welding speed)
V	- Volume
V_{arc}	- Arc voltage
VPPA	- Variable Polarity Plasma Arc
V_{ref}	- Required or reference voltage

1. EXECUTIVE SUMMARY

Objectives

The primary objective of this research was to design a control architecture for control of the Gas Tungsten Arc Welding (GTAW) process, and implement it in a prototype welding controller, to be delivered to NASA at the end of the research period. Specifically, some common problems, or potential problems, of the GTAW process were discussed with NASA and Rocketdyne personnel at the outset of the work, and the focus of the controller design was set on these specific items. The key issues addressed in the design of the delivered controller were: (i) enhanced control of the GTAW equipment parameters, and (ii) enhanced control of the Automatic Voltage Control (AVC) subsystem. These objectives were selected and approached with the welding tasks on the Space Shuttle Main Engine (SSME) in mind. Furthermore, all work and equipment selection was tailored to be adaptable to Variable Polarity Plasma Arc Welding (VPPAW) control at a later time, if desired.

Research Conducted

The research consisted of (i) review and study of existing approaches to enhanced arc welding control, (ii) research and conduction of experiments to examine the feasibility of possible solutions, and (iii) design and implementation of the selected techniques.

Extensive research of existing methods for arc welding modeling and control was carried out, and some of the key findings are discussed in this report. Numerous simulations and experiments were carried out to determine which techniques were viable and which should be rejected in the final implementation.

Selection of equipment and software items turned out to be less trivial than anticipated. A computer architecture based on the VME-bus, but using the Intel 80286 and 80386 processors, was finally selected. This approach combines the robustness of the industrial VME-bus and the software sophistication, economics, and availability of the MS-DOS/WINDOWS environment.

Research Conclusions

The main objective of the work, the design of a control system and its implementation in a process control computer, has been accomplished.

Potential Applications

Although the delivered welding control system is tailored for the GTAW process, other welding processes, such as VPPAW, can benefit from the use of the

system as well. Because the system functions as a generic welding controller, it has potential applications for commercial use as well as for NASA.

2. OBJECTIVES AND SCOPE OF THE WORK

The objectives of this investigation were originally stated in the Phase II NASA SBIR Contract Statement Of Work. These objectives have been slightly modified and adapted to immediate concerns relating to Space Shuttle Main Engine (SSME) Gas Tungsten Arc Welding (GTAW) at Marshall Space Flight Center (MSFC).

2.1 Introduction and Objectives

The overall objective of the SBIR Phase II research program was to develop a general architecture for an automatic GTAW system, tailored to SSME fabrication, and implement it in a welding controller.

At the outset of the research, numerous discussions were conducted with the NASA and Rocketdyne personnel involved with this work, and some specific areas of interest were determined. The general task of welding control was one, where it was considered desirable to improve the welder's overall control of the GTAW process. Acting on this objective, it was decided to build a prototype control system, utilizing computer technology to a larger extent than usually encountered in commercial systems.

The task of welding control was essentially twofold. First, the welding controller should aid the welder in selecting appropriate equipment parameters, such as voltage, current, travel speed, etc., for controlling the process. Secondly, the automatic voltage control (AVC) component of the welding setup was selected for substantial improvement. This was decided with the objective of attempting to minimize crater cracking, particularly during the tailout section of the weld. Most of the work conducted during the work period related to these two objectives to a more or less extent.

2.2 Results and Deliverables

The control system delivered to NASA consists of a VME-bus based computer, mounted in a standard 19 inch wide rack cabinet, with analog input and output ports for monitoring and control of the welding equipment variables. The hardware is of industrial grade, and thus well suited for use in the welding area.

The software reflects much of the work that has been done for this research, because software approaches were preferred over hardware approaches wherever possible. The decision to use software implementation to the greatest extent possible was based on the fact that software systems are very easily upgraded and improved. The digital AVC loop, implemented in the system, is an illustrating example of this. Earlier versions of the AVC were to a large extent hard wired in a similar way as most commercial AVC systems are. When work started on implementing

some of the more complicated features of this AVC, such as arc length modeling and control, frequent adjustments and changes in the control loop hardware became unacceptably difficult and cumbersome. In contrast, the entire structure of the software-based AVC can be tuned or altered through a few computer keystrokes. This flexibility has proved invaluable during the development of the system.

Microsoft Windows, version 3.0, provides the programming and user interface of the system. This environment became available relatively late in the research period, but nevertheless it was decided to use it for this work. This has turned out to be a worthwhile decision, as the user interface, with its menu and graphics capabilities, enhance the use of the system greatly.

3. BACKGROUND: EXISTING TECHNOLOGIES

This section serves to give an overview of some of the problems encountered in control and implementation of arc welding, and particularly the Gas Tungsten Arc Welding (GTAW) process. A number of techniques that have been used to solve some of these problems are discussed, and some new approaches are presented.

3.1 Areas For GTAW Process Improvements

The main purpose of this work was to examine ways to improve the GTAW process, and construct a prototype control system to demonstrate some of these methods. The approach taken here was to utilize computer technology to the fullest extent, and substantially beyond that of commercial off-the-shelf welding systems. The objective was to use the capabilities of the computer to improve some aspects of the process, which currently available welding equipment have not been able to do.

The general approach of welding control is one such area. Ideally the welder should be able to specify those parameters which characterize the finished weld, such as bead penetration or width, rather than equipment parameters (e.g., current, voltage, etc.) which indirectly and in a complicated manner determine the weld features. The use of mathematical models of the welding process is one approach to this. Furthermore, welding models can be utilized to simulate the welding process off line, and thus aid the welder in getting acquainted with the effects of individual equipment variables and help him select appropriate welding parameters prior to welding. A number of weld models have been studied during the research period, and they are discussed in this report. Three specific models have been selected for implementation in the delivered system.

A specific control problem which has been addressed in the research is the Automatic Voltage Control (AVC). The traditional AVC system is an analog electronic system, which has been determined to be rather limited in its capabilities to operate properly for arc voltage control. The deficiencies of conventional AVC systems have long been known by welders, and one consequence of these shortcomings is that AVCs must be disables when welding parameter values, particularly current, are varied over wide ranges. This applies in particular to downsloping of current as the weld is terminated. A digital AVC implementation permits real-time adjustment of loop gain (and therefore the AVC damping factor), and numerous other real-time adjustments which enhance the AVC operation. In addition to adjustment of the system damping factor, the digital AVC offers true arc length control through modeling of the arc length as a function of arc voltage and arc current. The digital AVC is implemented in the system for NASA.

Finally, recording of welding parameters is implemented through two different approaches: (i) a computer database, and (ii) a real-time weld parameter recorder, with extensive analysis facilities. All of these features are discussed at length in this report.

3.2 Weld Modeling

Weld modeling, i.e., prediction of the weld pool shape or other characteristics in terms of given equipment parameters, plays an important role in the system design. An adequate model provides the welder with estimated information about the weld, prior to actual welding. Furthermore, a welding model can be used to determine the equipment parameters required to achieve certain weld characteristics, such as penetration, bead width, etc.

Modeling research to this date has concentrated on two different approaches: (i) model derivation from the physics of the welding process, and (ii) empirical model derivation, based on data obtained from the welding process. In some cases, models reflect both approaches to some extent. For example, physics-derived models may include various adjustment factors to improve the agreement of the model with real welding data. Similarly, models largely based on welding data are usually composed based on some knowledge of the physics of the process. Various modeling approaches are exemplified in the following paragraphs, where a number of existing weld models are discussed.

The parameters of primary interest in this research for description of the weld bead geometry were bead width, penetration, reinforcement height, and bead cross sectional area (refer to Figure 1). These parameters are predicted, or estimated, by a weld model. The equipment parameters which control the geometry of the weld pool include arc current, voltage, travel speed, and wire feed rate. These two sets of parameters are referred to as Direct Weld Parameters (DWPs) and Indirect Weld Parameters (IWPs), respectively.

Multiple regression models were developed by Hunter, Bryce, and Doherty [1]. They performed numerous welding experiments and suggested using polynomials to relate the observed weld parameters to equipment parameters. Their work was done with GMAW but the inherent method is just as well applicable to GTAW. For a general weld parameter P (which may represent bead width, penetration, reinforcement, etc.) they proposed the following model:

$$P = A + B\, S^{c1}\, W^{c2}\, V^{c3}\, E^{c4} \qquad (1)$$

where A, B, and c1 through c4 are empirical constants. S, W, V, and E are travel speed, wire feed rate, voltage, and contact-tip-to-work distance respectively. Such

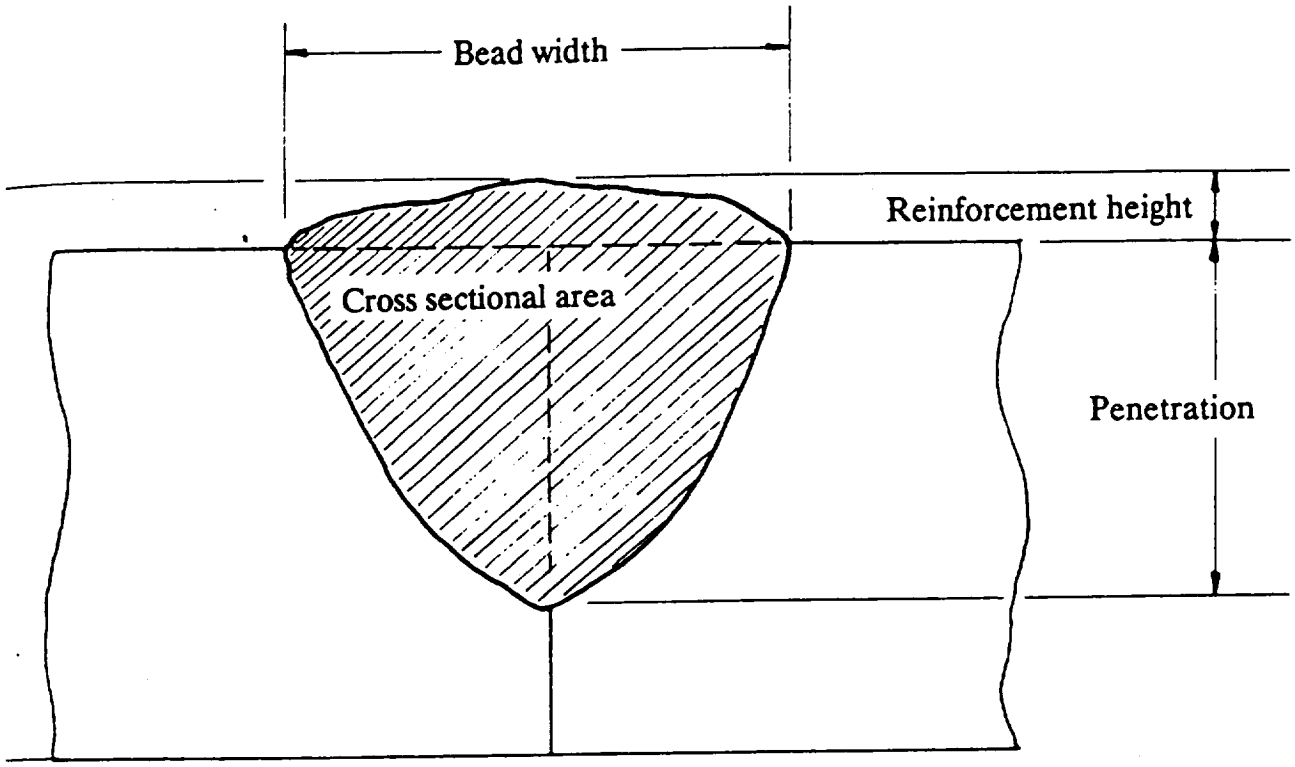


Figure 1. Examples of direct weld parameters (DWP).

multiple regression, approximating each weld parameter as a polynomial in the equipment parameters, is a generally valid approach. But since the forms of the mathematical approximations are in no way related to physical processes these approximations are only valid for narrow ranges around specific operating points.

Smartt and Key [2] conducted a series of welding experiments to establish the factors controlling bead geometries. They made single pass partial penetration welds on stainless steel, with and without filler metal. The voltages ranged from 9.0 to 10.5V, the current was 100 to 200A, and the welding speed varied approximately from 1 to 4 mm/sec (2.4 to 9.4 ipm). Where filler wire was used its diameter was 1.14 mm (0.045") and the wire speed ranged from about 2 to 17 mm/sec (4.7 to 40 ipm). Detailed numerical results are given in the reference.

In summary, bead-on-plate welds *without filler wire* generally resulted in an increasing width when the heat input per unit length of the weld was increased, as expected. A different behavior was seen for penetration, which first increases as well, but then starts decreasing again beyond a certain current level, typically around 150A. Experimental data for welds *with filler wire* did not yield a conclusive general trend. The effects of wire speed on bead width were not shown explicitly but increased wire speed decreased penetration, and added to the reinforcement as one would expect. Finally, experimental data revealed that the penetration/bead width ratio is about 1/2 only for welds without filler and only for reasonably high welding speeds. Otherwise the ratio is smaller.

Smartt and Key did not attempt to derive a mathematical model directly from the vast amount of data they produced. On the other hand they presented a rather simple model of the weld pool, assuming that its shape is hemispherical, where they derived the radius of the hemisphere as a function of power. They were able to fit this model reasonably well to some of their experimental data.

Principally this model is too rudimentary for controlling a general welding procedure. It does not account for the effects of moving the welding point along the welded surface, finite dimensions of the weldment, the effects of filler wire; and the assumption of a hemispherical pool shape is rather crude. However, the model may be applicable for quick approximate estimates of weld pool dimensions, while more sophisticated models, which take more time for calculations, may be used to obtain more accurate estimates.

Dornfeld, Tomizuka and Langari [3] set out to determine which parameters of the welding equipment and other affecting factors control the weld characteristics and bead geometries, as well as defects. They pointed out that the final properties of the weld are determined by the dynamics of the weld pool and the equipment, and therefore static relationships were of limited value. Although they used Gas Metal Arc Welding (GMAW) as a representative welding process the general ideas and

much of the specific conclusions can be applied to our task of analyzing the GTAW process. Without citing specific references or tests they collected a set of weld characteristics representing weld quality in general, and the parameters which control or affect them. The results are summarized in Table 1. Although the table displays qualitatively the factors determining weld characteristics no quantitative information is provided.

Hardt, Garlow, and Weinert [4] proposed a dynamic model for the bead width of full penetration welds (refer to Figure 2). The weld pool was modeled as a cylinder of height z equal to the thickness of the welded plate and radius r. Their model is founded on the following relationship:

$$Q_{in} = 2\pi rkz\,(dT/dr) + h\,(dV/dt)\rho \qquad (2)$$

where k is thermal conductivity and h is the fusion heat of the material, T is the temperature at the pool sidewall and V is the pool volume. The equation simply states that the total input power equals power lost through sidewall conduction plus power absorbed to melt additional metal and expand the pool radius. The above equation is expanded into the following form:

$$Q_{in} = \{2\pi kz(dT/dr)\}\,r + \{2\pi r\rho hz\}\,dr/dt \qquad (3)$$

The first term of the sum can be interpreted as a conductive component and the second term represents capacitance. Note that both terms exhibit nonlinearity, as r occurs within both of the bracketed expressions. Hardt et. al. took the approach of assuming a nominal value of r, and thus treat the bracketed terms as constants. Then they solved this linear first order equation and obtained pool radius as a function of time and input current I, noting that $Q_{in} = EI$ (for GTAW the voltage E is to a first approximation constant as long as the arc length is held constant). For fixed operating points they obtained the steady state value and the time constant of the bead radius for given current inputs. Additionally, assuming that the boundary temperature gradient dT/dr is proportional to the welding speed v, they arrived at and verified similar solutions for bead radius as a function of the inverse of the welding speed, 1/v. The results obtained from the model correlated well with experimental data for the conditions they used.This model is interesting for the sake of its dynamic capabilities. The radius r of the weld puddle (which is assumed to be cylindrical) is derived by solving the differential equation:

$$nE(t)i(t) = C(r)\,dr/dt + G(r)\,r \qquad (4)$$

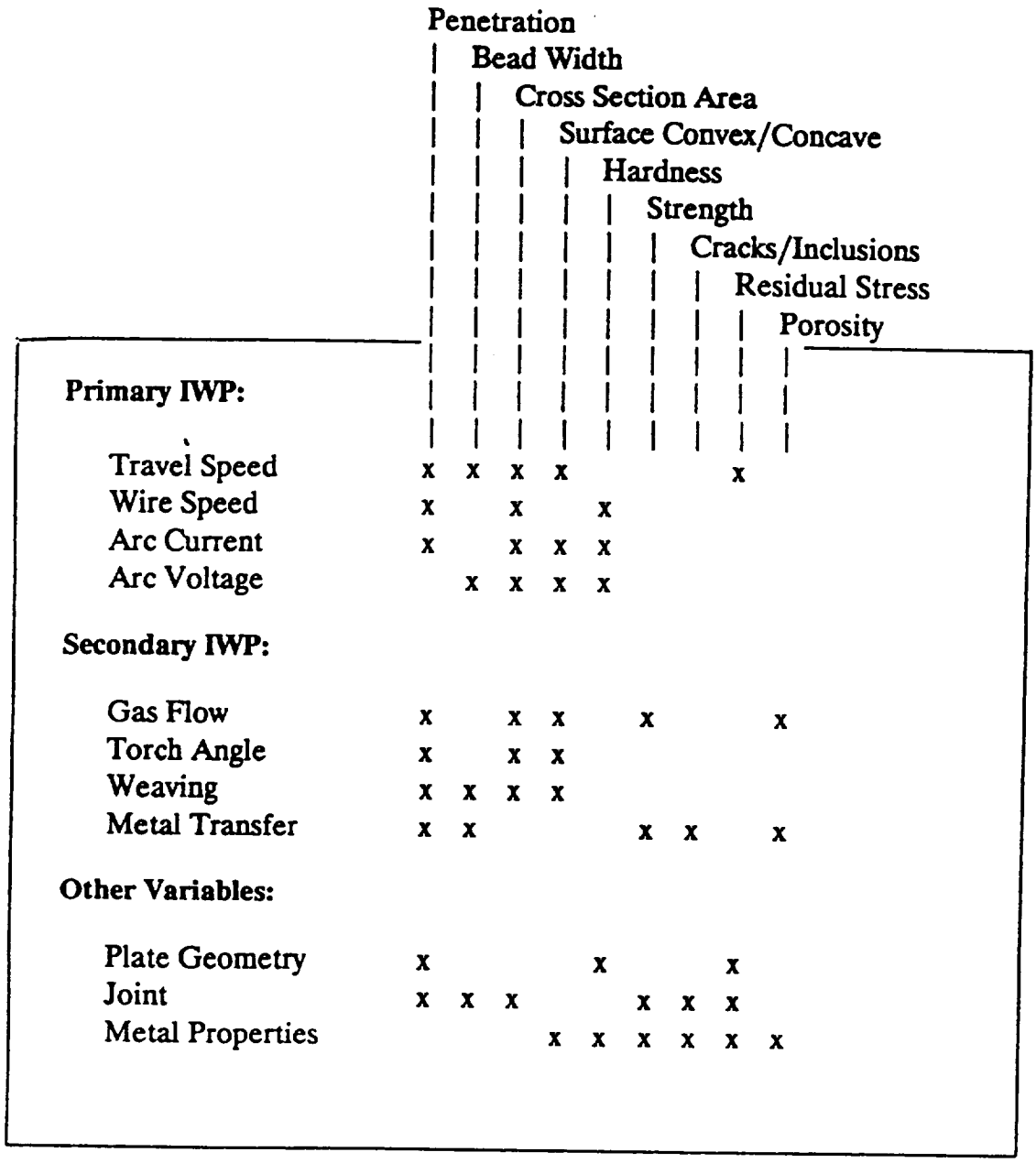
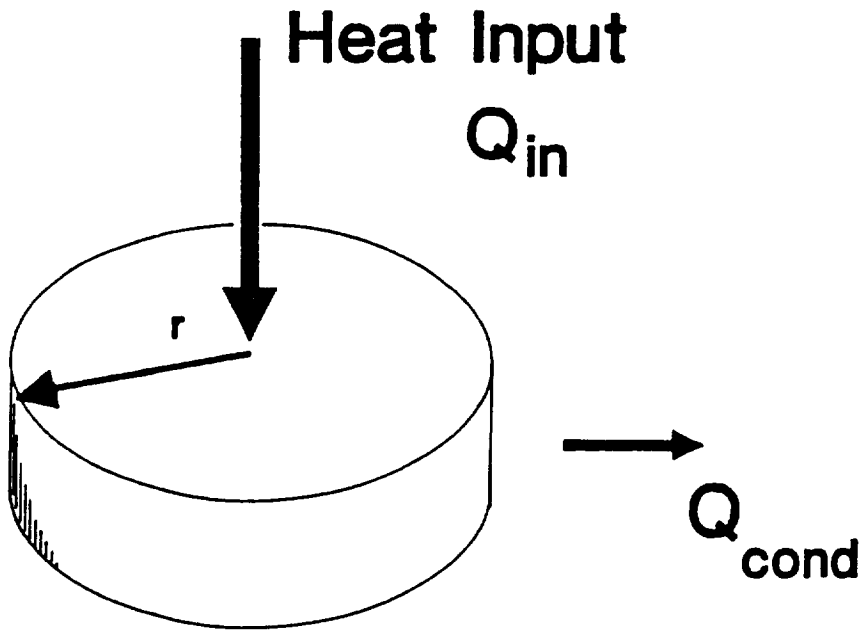


Table 1. Common direct weld parameters (DWP) and typical indirect weld parameters (IWP) contributing to them.



$$Q_{in} = Q_{cond} + (rho)\ h\ \frac{dV}{dt}$$

Figure 2. A dynamic cylindrical weld pool model.

where n, E(t), and i(t) are the efficiency, voltage, and current respectively. The quantities C(r) and G(r) can be viewed as "capacitance" and "conductance" (refer to Ohm's law for parallel RC electrical circuits: $i = C\,dv/dt + G\,v$). In this case C and G are both functions of the radius r, in addition to material constants, and this complicates an analytical solution of the system. Hardt et al. assumed constant values for C and G and examined the first order time constants and steady state values of r based on the model.

The above models exemplify the various approaches to weld modeling that were examined during this research. Due to oversimplifications and inaccuracies, however, none of these models were deemed feasible for implementation in the system constructed during this work. The modeling approaches implemented here are discussed separately in Section 4 of this report.

3.3 Welding Control

The welding control system is a multivariable control system wherein the decoupling of the various inputs and outputs has been identified as being an extremely difficult task. Further, the knowledge of the behavior of the systems has been found to be heuristic to a large extent and extremely sensitive to the region of operation. Thus it is imperative that adaptive controllers must be used. The choice of controllers may vary from one operating situation to another. Thus it must be possible for the same system to have a variety of controllers which may be used in the different regions of operation. This means that the system must be adaptive or, at least be able to effectively control highly nonlinear and ill-defined processes. To a welder such information regarding the system behavior may not be available. Thus the following options must be provided:

1. As a first level estimate, the system must be able to suggest the required equipment parameters. This may be achieved using a combination of heuristic rules, models that predict the same, and expert knowledge gained with the use of the system to perform the actual welding operations.

2. The system must be able to decide the kind of controller that could be used depending on the region of operation. This implies that the selection and supervision of the control algorithms are done by the system.

3. The user must be informed about the current state of the system in terms comprehensible to him and which help him decide on the course of action.

4. The system must also possess some intelligence to ensure quality control [5]. The various available sensor technologies may be able to measure the physical quantities rather accurately but the interpretation of these measurements must ultimately be done by an expert user which may not be feasible at all times. Hence the system must be able to interpret the sensor values (maybe both off and on-line) and suggest the possible defective areas and guide the user to look for possible defects [6, 7].

A number of methodologies have been employed by a number of researchers to control the arc welding process. Some of these were discussed in the Phase I Final Report preceding this Phase II effort. These methodologies varied from classical linear control approaches to expert system techniques [8]. Sliwinski and Ruokangas [9] of Rocketdyne outlined a prototype GTAW system for welding the SSME. Although their emphasis was largely on the robotic automation they provided a good overview of the general process. They proposed a weld parameter data base to provide an initial set of process parameters for each specific weld. This data base holds the equipment parameters, originally obtained by detailed testing and optimization. A second section of their system consists of a sensor controller which principally processes and communicates sensor data to other system components. Finally a custom user interface was proposed.

The primary candidate selected here for the task of parameter control during this research is the artificial neural network. The neural network principles are discussed in further details in Chapter 4, but here it is emphasized that such networks have proved to be amazingly successful at emulating complex, nonlinear, multivariable processes. Examples and numerical results, illustrating the performance of neural networks in this respect, are presented in Section 4.4.

A variety of sensors exist for monitoring the weld process [10]. Whenever reasonably applicable, direct monitoring of the process is preferable to estimates using models or other indirect methods. Sensors for monitoring weld geometries in real-time [11] are of primary concern in this research. Therefore explicit off-line sensors (e.g. for quality control) and tracking sensors (for robotic manipulator guiding) will be mostly ignored. Furthermore it should be noted that this overview of sensors is presented for the sake of completeness as much as relevance to our final architecture proposal. Our conclusions from this research effort reveal that an advanced welding system does not have to rely on a vast amount of sensors, but rather should utilize a few well chosen ones and process the information they yield to give the required and relevant information about the welding process.

The Through-The-Torch-Vision (TTTV) System, developed at Ohio State University [12], is currently applied at NASA. A bundle of optical fibers is run from

the welding electrode up to a camera which thus is able to record the immediate vicinity of the electrode. The recorded image is digitized and processed by a computer. Its use so far has primarily been for seam tracking, while monitoring of bead width and surface conditions may be performed by this or a similar system.

For penetration monitoring at least two sensors have been tested at Rocketdyne. One is a bead contour monitor [13], directing a laser beam to the molten pool surface and receiving the reflection, whose direction is controlled by the contour of the pool surface. The reflection direction is recorded and the convexity of the reflecting surface interpreted as a measure of penetration. Another sensor for penetration works with the laser beam contour monitor and interprets acoustic emission received from the weld pool [14]. The acoustic resonance frequencies of the weld pool are determined by its dimensions and by the reflectivity of its boundaries. The pool of a partial penetration weld is totally supported by a solid metal bottom which opens up when full penetration is reached. Penetration information is obtained by monitoring and interpreting the frequency components of the acoustic signals from the pool. Used together these two penetration sensors monitor when the weld is under-penetrated, over-penetrated, or acceptable. Salter and Deam [15] and Deam [16].

Infrared thermography (Chin et al [17]) is a powerful but expensive means of monitoring the temperature profiles of the workpiece during welding. A specialized camera, sensitive to wavelengths in the infrared region of the spectrum (around 2 to 6 microns) is used to examine temperature distributions as welding proceeds. This reveals important information, such as heating and cooling rates at specific points, and isotherms around the weld puddle and into the heat affected zone. Besides temperature information for metallurgical purposes the symmetry of the isotherms on the workpiece is an indicator of correct seam tracking. Discontinuities in the metal can also cause distortions in the isotherm patterns. Therefore the infrared thermography technology is a powerful tool for monitoring the weld process in real-time. Work has been done by General Digital Industries, Inc. to incorporate infrared sensing in a variable polarity plasma arc welding (VPPA) controller being developed for NASA.

3.4 Data Recording

Means for recording welding data have been studied, and two approaches are considered here.

The first approach is using a database to keep records of welds as they are carried out. Such a database must facilitate relatively easy retrieval of welding records, and similarly the database must be easy in use for storage of new welding parameter records. Some of the key parameters relating to the base material to be stored in a GTAW database are base material type (metal), some dimensional in-

formation (e.g., thickness), joint type, groove preparation. The primary welding parameters, used for real-time control of the weld, are arc current, arc voltage or length, travel speed, and wire feed rate. Additionally, the wire material and diameter, electrode type, size and tip angle, and gas composition and flow rate must be recorded as well. Such a database can be used as a check or tool for comparison with the welding parameters suggested by the model-based welding controller.

A second approach to records keeping is to use a computer to sample the applied welding parameters in real-time, as the weld is carried out, and store these parameters in a computer file. Such recordkeeping reveals more details of the actual process, and can offer the welder various means for monitoring the weld in real-time and analyzing it at a later time. Such analysis may use various statistical tools, such as calculation of average values, standard or maximum deviations from nominal values, trend analysis, etc. Furthermore, the results may be displayed graphically, and thereby give the welder a better view of how the weld proceeded than by merely looking at numerical results.

Both of the above approaches are implemented in the system that is delivered to NASA as a part of this work.

4. WELD MODELING

The physical dimensions of the molten pool are major factors in determining the structural adequacy of the weld, primarily its strength. Such factors, which characterize the finished weld, are referred to here as *direct weld parameters (DWP)* [18]. They include the penetration of the weld pool, the bead width, the transverse cross sectional area, and the height of the reinforcement. These direct weld parameters are governed by numerous factors, e.g., welding voltage, current, torch travel speed, wire feed rate, electrode tip angle, and shielding gas type and flow rate. Collectively they are referred to here as *indirect weld parameters (IWP).* The task of the human welder is to select and control the IWPs in such a way as to obtain some desired DWP values. For critical applications this usually requires substantial experimentation. Furthermore, not all output variables can be observed in real-time. While, e.g., beadwidth can be monitored while welding is in progress, the pool penetration is usually unknown on-line. A practical approach to estimating DWPs that cannot be sensed is use of a reliable model of the weld pool describing the pool geometry in terms of the heat input and other controllable parameters. This is equivalent to a mapping from indirect weld parameters to direct weld parameters.

Models of the weld pool are usually based on the physics of the weld or empirical data. Frequently models combine both approaches to some extent; for example physics-derived models often include "efficiency" or other parameters which may be adjusted to fit the model to actual welding data. An alternative classification of weld models defines them as either *static* or *dynamic*. The static models describe the relations between process variables after they have reached equilibrium while the dynamic models provide additional information about the process during transitions. In addition to welding-specific models, general finite-element algorithms provide relatively accurate information about the weld pool. One numerical model based on discrete-element techniques is the one developed by Eraslan et al [19]. Although this and other models using temporal and spatial discretization are more general and frequently more accurate than the application-specific models listed earlier they usually require far more computations. Therefore they are better suited for off-line simulations than real-time control purposes.

A significant amount of the work performed for this research consisted of accumulation of information on previous work in the field of weld modeling and evaluation of the various methods. Examining mathematical models of the weld process constituted a large amount of the total research efforts. It was believed to be justifiable, as the models provide:

a) information on which equipment parameters affect a given weld parameter and how they are related;

b) estimates of weld parameters which otherwise can not be measured in real time;

c) general insight into the behavior of the weld process, aiding the system designer in the general design.

This section summarizes some of the most important findings from the modeling research.

4.1 Review

A number of models of the weld pool have been developed by various researchers. Steady state analysis of the welding processes has dominated much of this work. Some of these models have already been discussed in this report. In this section emphasis is put on the classes of models that have been implemented for this research, i.e., physics-based models, derived using similar methodology as Rosenthal used for his model, and the empirical neural network model.

The models of the pool geometry can be used to determine the setpoints of the welding equipment, such as welding current, voltage, wire feed rate, and travel speed. Two primary parameters of the weld cross section which commonly are used to determine its geometry are bead width and penetration. Additional factors of interest include the reinforcement area, the total cross sectional area, measures of symmetry, and other information about the bead shape. The bead width, penetration, cross sectional area, and reinforcement height, have been used throughout this research as the key geometrical parameters for model analysis.

The Rosenthal model [20] was the first analytical solution of heat flow in welding, presented in 1941. The partial differential equation

$$\nabla^2 T = (1/\alpha)\, dT/dt \qquad (5)$$

describing the temperature field T(x,y,z) was solved for a point source of a constant value, moving at a constant speed v in the x direction of a Cartesian coordinate system attached to the point source. The temperature at any given point (x,y,z) with respect to the heat source was derived as

$$T = T_0 + (Pv/4\,\alpha^2\rho c)\exp(v(vt-x)/2\alpha + r) \qquad (6)$$

where T_0 is the initial temperature, P is the welding power, α is the thermal diffusivity, ρ is the material density, c is the specific heat, and r is the distance between the point of measurement and the point source. The steady state nature of this

model comes from the assumption that the temperature distribution does not change as viewed from a coordinate system following the moving source.

In short, Rosenthal's model provides temperature values for any point in the welded plate, as a function of material constants, plate thickness, welding speed, initial temperature, and the rate of heat input (power). By iteratively applying the model, any isotherm can be located in the metal plate. Thus an estimate of the steady state pool surface boundary, where the temperature equals the melting temperature for the base material, can be found. This model suffers from several deficiencies introduced by the simplifying assumptions. The weld is presumed to be made on an infinitely large plate as measured in the x and y dimensions. Therefore heat is assumed to be freely conducted away from the welding region. The metal characteristics are assumed to be constant throughout the weld pool and the entire plate volume, and convection currents in the molten weld pool and the effects of filler wire are ignored, as are phase changes of the metal as it is melted and frozen again along the welding track. Typically the results of this model correlate rather poorly with experimental results, with deviations on the order of several hundreds of per cent. Considering this, Rosenthal's model is not a feasible candidate for a weld control system.

A.C. Nunes [21] proposed an extension of the classical Rosenthal model. He derived a multipole expansion of the moving power source and used these multipoles to represent physical phenomena not otherwise taken into account by Rosenthal's model nor most other weld models. Studies (cf. Mills [22]) have revealed that the liquid metal flow patterns in the weld pool are of primary importance and that arc current density and the base metal properties have secondary effects on weld geometry. Nunes introduced a dipole located at the point source and oriented along the welding direction. The end of the dipole in the direction of movement was a heat sink representing absorption of heat by the metal as it melts along the track, while the other dipole end was a heat source which delivered heat back to the metal slab as the molten pool solidified. Three quadrupoles, aligned along the x, y, and z axes, were introduced, as well, to replicate convective circulations in the pool.

Yet another extension of Rosenthal's model has been suggested by N. Tsai [23]. Rather than expanding the heat source as dipoles and quadrupoles, as Nunes did, Tsai took the approach of representing the heat source as a disk, delivering maximum energy at the center. Specifically he assumed a cylindrical gaussian distribution of heat flow, as depicted in Figure 3. Contrasted to the required knowledge of monopole, dipole, line, and quadrupole strengths of Nunes' model this representation requires specification of the total power as well and the standard deviation of the gaussian distribution for defining the weld puddle.

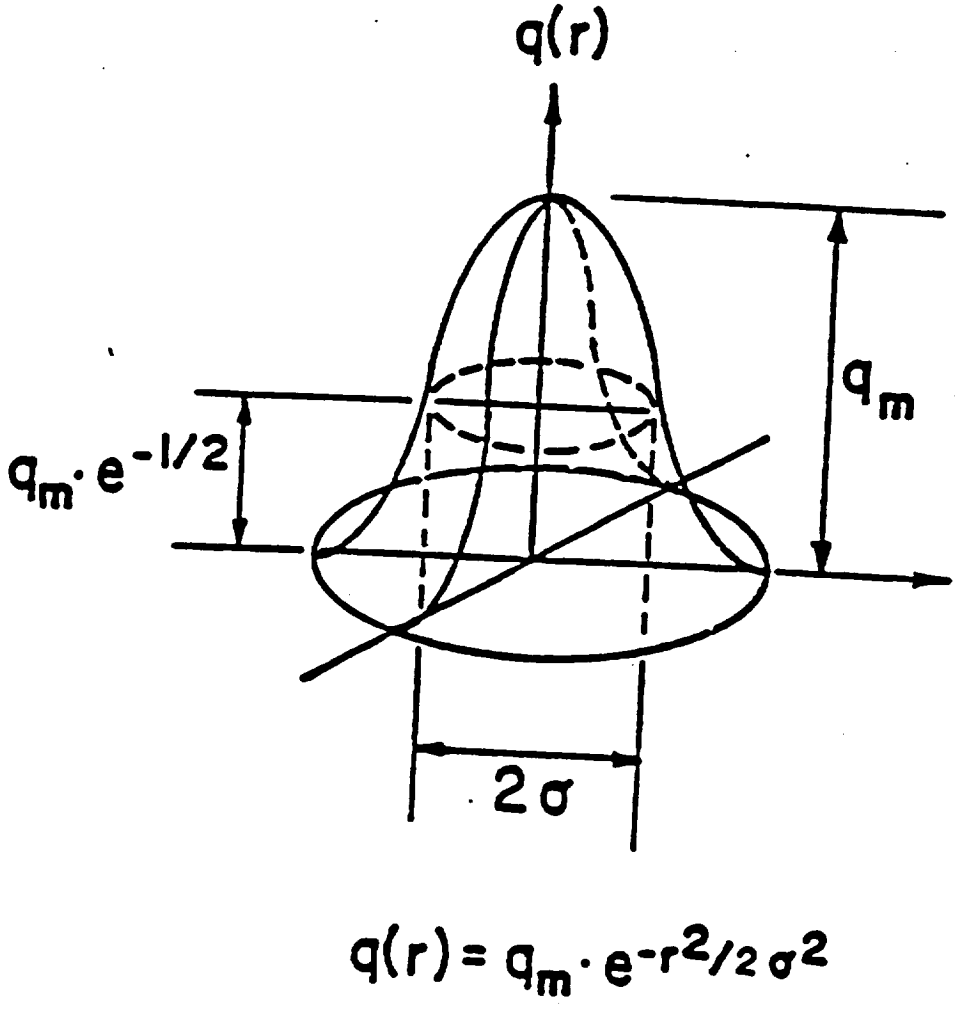


Figure 3. Gaussian heat input distribution: Tsai's model.

4.2 Nunes' Model

This model is a modification of the model devised by Rosenthal. A multipole expansion of the moving power source is derived and these multipoles are used to represent physical phenomena not otherwise taken into account by Rosenthal's model. This model estimates the shape of the GTAW pool during steady-state welding, i.e., welding current, voltage, travel speed, and other parameters are assumed to be constant and any transients in the system are assumed to have ceased.

4.2.1 Underlying Theory Of Nunes' Model

Nunes' model estimates the shape of the GTAW pool during steady-state welding, i.e., welding current, voltage, travel speed, and other parameters are assumed to be constant. The model is derived from the mechanisms of heat conduction in the welded metal. It accounts for the effects of travel speed and it models the workpiece as an infinitely large plate of finite thickness. The effects of feed wire is not accounted for explicitly (although it may be modeled to some extent through the multipoles, which are discussed below).

An orthogonal coordinate system is assigned to the moving torch (see Figure 4). Its origin is at the workpiece surface, in the middle of the arc. The x-axis is directed horizontally along the travel direction, the y-axis is directed horizontally to the right with respect to the torch which moves forward, and the z-axis is directed vertically down into the workpiece. As the torch, and the molten pool, move along the workpieces surface this coordinate system follows along.

Unlike some less advanced models, derived from the same physical principles, and which model the source of heat simply as a point source, this model uses additional and more sophisticated forms of heat sources as well. First, the fact that the metal is constantly being melted at the front of the moving arc (absorbing latent heat), and re-solidified at the trailing edge (returning the latent heat), is accounted for by the use of a point source and a point sink of heat in the pool. This source and sink are referred to as the model *dipole*, which is oriented along the axis of torch travel. The dipole acts as to shrink the pool contour at the front of the arc and expand it somewhat at the back. The model user can specify the intensity of the dipole in the model. An informal view of the effect of the dipole intensity is to relate it to the torch travel speed. When the speed is zero, the concepts of melting at the arc front and solidification at the back do not exist, and correspondingly the dipole value is zero. As the travel speed is increased the effects of melting at the front and solidification at the back increase. The user can model this by increasing the dipole intensity.

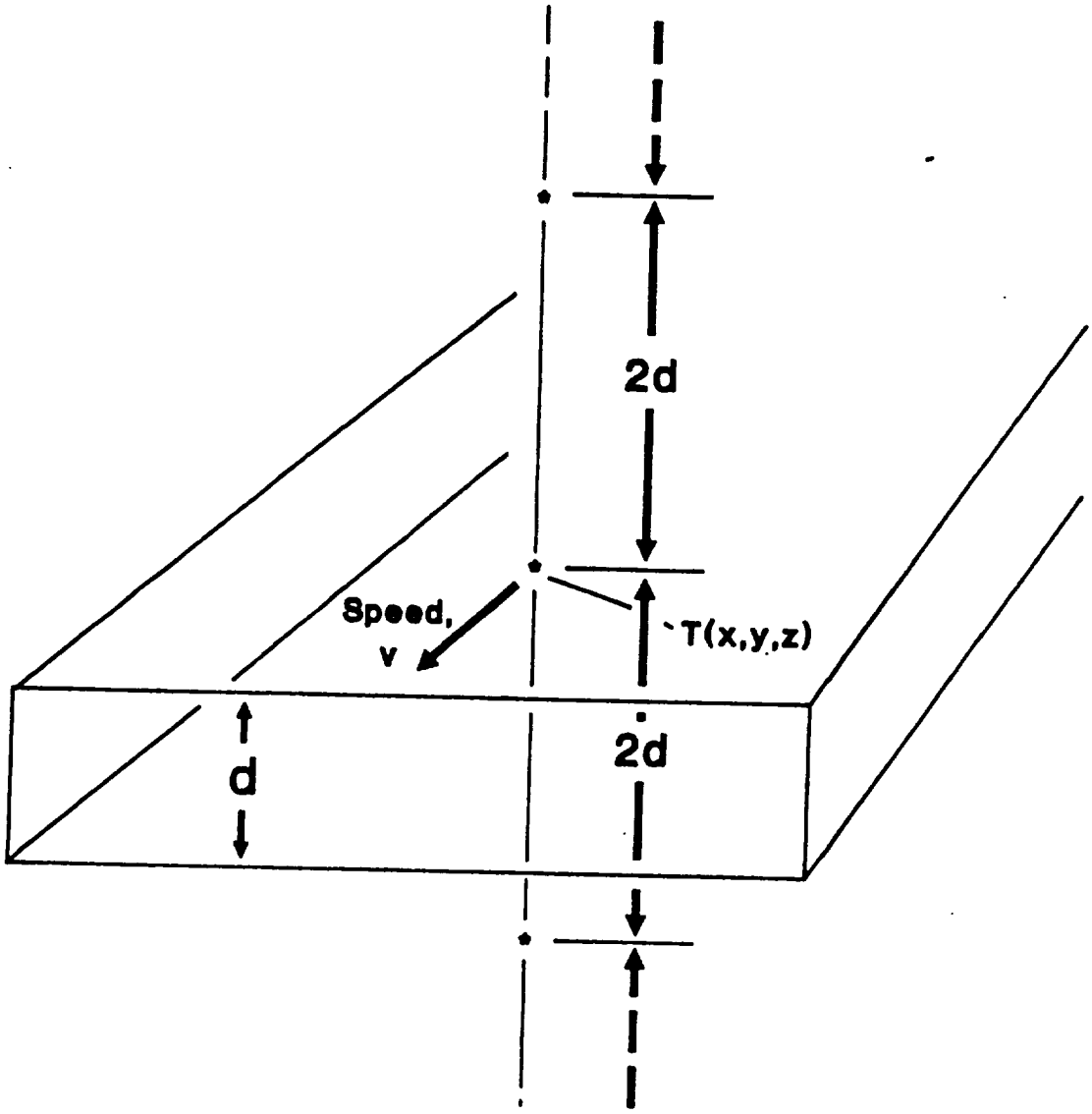


Figure 4. Nunes' model and its coordinate system.

A second form of combined heat sources/sinks is that of *quadrupoles*. The model allows the user to specify three different quadrupoles, referred to as *quadrupole x*, *quadrupole y*, and *quadrupole z*. Quadrupole x consists of two heat sources and two heat sinks, all located on the x-axis at the origin. This quadrupole models the transfer of heat at the pool surface, along the axis of travel, due to circulation of molten metal in the pool. Similarly, quadrupoles y and z model transverse circulation and heat transfer from the pool bottom and up to the pool surface, respectively.

In summary, the combined effects of the dipole and the quadrupoles affect the shape of the pool. With all of these four multipoles and the travel speed equal to zero, the pool will be shaped as a hemisphere. Increasing the dipole intensity will shrink the pool at the front and elongate it at the tail. A somewhat similar effect is obtained by increasing the travel speed. Increasing the x-quadrupole intensity generally elongates the entire pool. Increasing the y-quadrupole widens the pool, and increasing the z-quadrupole deepens the pool. The units of the dipole and quadrupoles are KW inches and KW inches^2, respectively. Although this may at first appear as additional power applied to the pool, it should be recalled that each of the multipoles contains an equal number of heat sinks and heat sources and thus the net power contribution of the dipole and the quadrupoles is zero. Therefore, these figures are not indicative of energy added to or extracted from the pool, but rather energy redistributed within the pool.

Nunes' model provides, in essence, the temperature at any given point in the weldment, based on seven power source variables and the welding speed. These power source variables are:

Arc Efficiency:	The fraction of the power source output which actually enters the weldment surface and contributes to melting of the metal.
Line Contribution:	The fraction of the surface power which is modeled as a line heat source, aligned along the axis of the arc. For most GTAW purposes, as contrasted to laser or electron beam welding, this can be assumed to be zero.
Point Power Source:	The amount of power modeled as a point heat source.

Dipole Component:	The amount of power used to melt the weld puddle front and the power delivered back as the metal solidifies.
Quadrupole Components:	Power values indicative of energy circulations within the weld puddle. There are three orthogonal quadrupole power elements.

Varying these seven parameters, one can obtain a wide variety of pool geometries for a fixed travel speed. Compared with less sophisticated models, which assume some simplistic pool shapes (e.g. hemispherical) or which do not provide as many controlling inputs (such as the original Rosenthal model), this model is superior. On the other hand, there are a variety of other models providing a number of controlling parameters, such as various regression approximations. In contrast to most of these, the controlling parameters of Nunes' model are relatively more effective and transparent to the user, as they can be interpreted in terms of underlying physical phenomena (phase changes, circulations, etc.).

Obtaining the temperatures at arbitrary points is the key to obtaining the simulated weld geometry. The boundary surface of the weld within the plate is obtained by traversing from an initial boundary point so that the calculated temperature equals the melting temperature. The initial boundary point on the weld pool is found by traversing along the surface of the weld plate, away from the center of the arc. When the modeled temperature reaches the melting temperature for the metal, this point has been found. Successive points on the weld pool boundary are found by searching around the previous point, until another point is found at the melting temperature, and so on.

4.2.2 Nunes' Model Implementation

For this research, Nunes' model has been implemented in three versions, with each one utilizing improvements derived from previous versions.

The first implementation was written in the C-language, on a Hewlett-Packard 9000/300 workstation. This model was largely based on an earlier Basic code (originally implemented at NASA-MSFC) and a Fortran code (developed during the Phase I period of this research). The workstation implementation used the X-Windows graphics library, which facilitated window and menu-based operations. A typical screen from this implementation is shown in Figure 5. The model parameters and other variables are listed in the Parameter List, which can hold up to 3 different parameter sets and the corresponding cross sections calculated by the model are plotted in the large window, designated as Graph 1. In this specific case

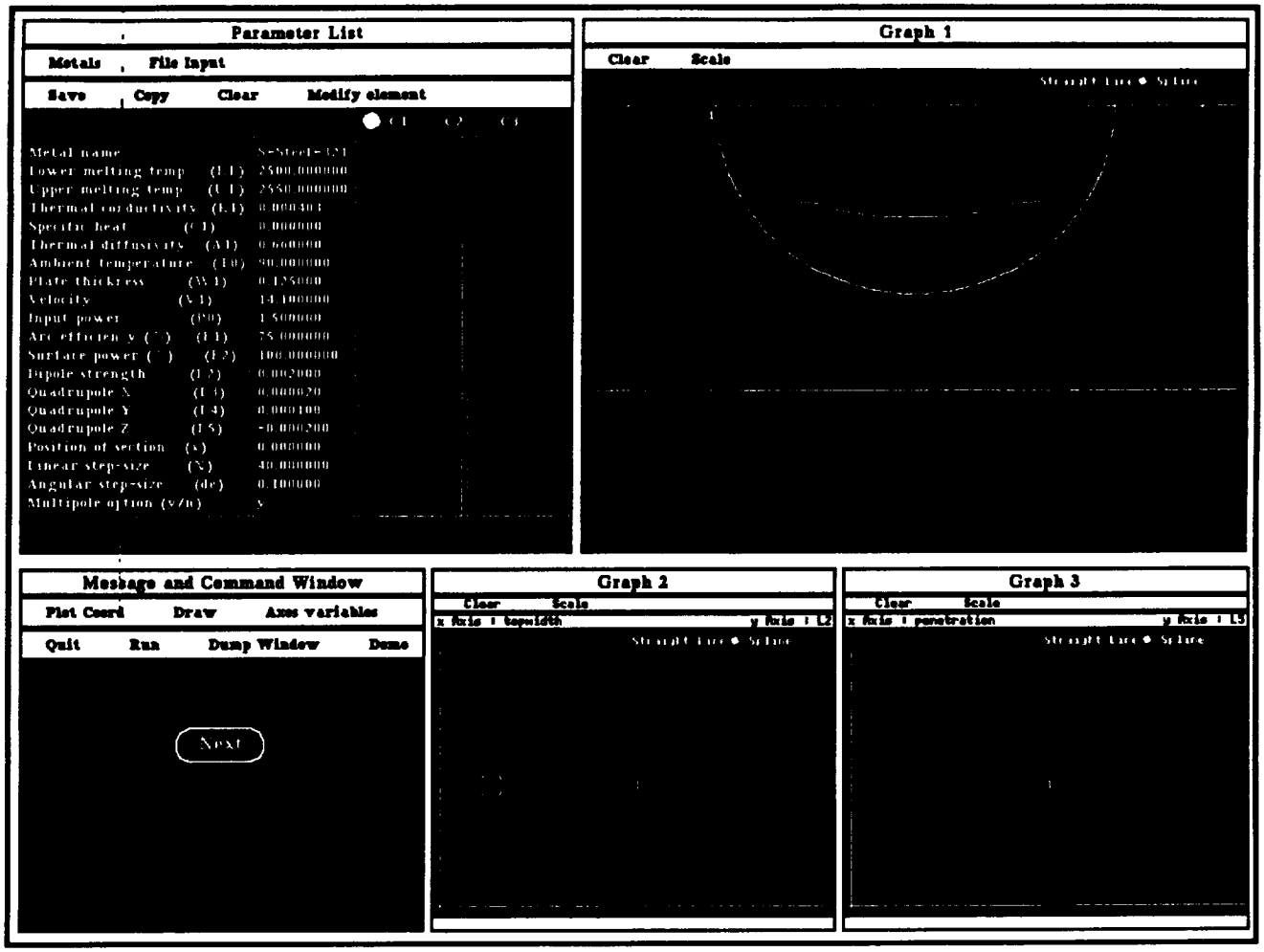


Figure 5. Transverse cross section of an actual weld bead and a bead cross section simulated by Nunes' model (marked #1). The actual cross section was entered point by point while the simulated one was calculated using the parameters shown in the parameter list column (C1).

the parameter sets are all relatively similar and thus the 3 model plots coincide to a large extent. Additionally, cross sectional data from an actual weld can be used to plot the fourth curve in this window as shown. Graph 2 and Graph 3 show top width vs. dipole strength and penetration vs. z-quadrupole strength, respectively, for the three model sets. Any one parameter can be examined as a function of any other parameter in the Graph 2 and Graph 3 windows. This is useful in determining if one parameter can be expressed as a simple (e.g., linear) function of another parameter. The Message And Command Window is used for execution of the program and related tasks.

Representative results showing typical model comparisons to real welds are shown in Figures 6 through 8. These particular welds were made with DC current of 100, 150, and 200A, respectively. The arc length was 0.040 in. (resulting in respective voltages of 7.20, 7.50, and 8.14V), the travel speed was 10 ipm, the 2% thoriated tungsten electrode tip angle was 37 degrees, and the Argon shielding gas flow was 25 cfh. The welds were made on 1020 mild steel plates (2 in. by 12 in., 0.1875 in. thick). In all cases the actual weld cross section is labeled as "1", while the modeled cross section is denoted as "2". The cross sections of the actual welds were obtained by casting the weld crater impressions and obtaining the projection silhouettes as described later in this section.

The workstation implementation of the model was rewritten for MS-DOS computers, where the X-Windows interface was replaced by one developed with the dBase database program. This version of the model was delivered to NASA for evaluation, and it was demonstrated to NASA personnel during a visit to MSFC. In developing the MS-DOS version of the model, significant changes were made to the program in three areas. First, it was necessary to revise the code and reprogram all sections which used instructions specific to the Hewlett-Packard programming environment. This applied specifically to all input routines and graphics, which were designed around the X-Windows system on the Hewlett Packard workstation. Secondly, the opportunity was taken to further modularize the code in the MS-DOS version was taken. Specifically, the theoretical model was implemented by itself, independent of sophisticated graphics and input facilities, resulting in a set of self-supporting code modules. This part of the code was written to be compatible with the new ANSI C standard, which should make future re-implementations of the program under other operating systems more readily accomplished. Separately from the theoretical model kernel, the graphics routines were implemented in MS-DOS-specific code segments. Future implementations for other operating systems will require changes in this part of the program, but it should be significantly simpler now that it has been separated from the model kernel. The C-compiler from Microsoft, Inc. was used to create the executable program. Finally, a menu-driven database environment was implemented to enter the parameters to the model.

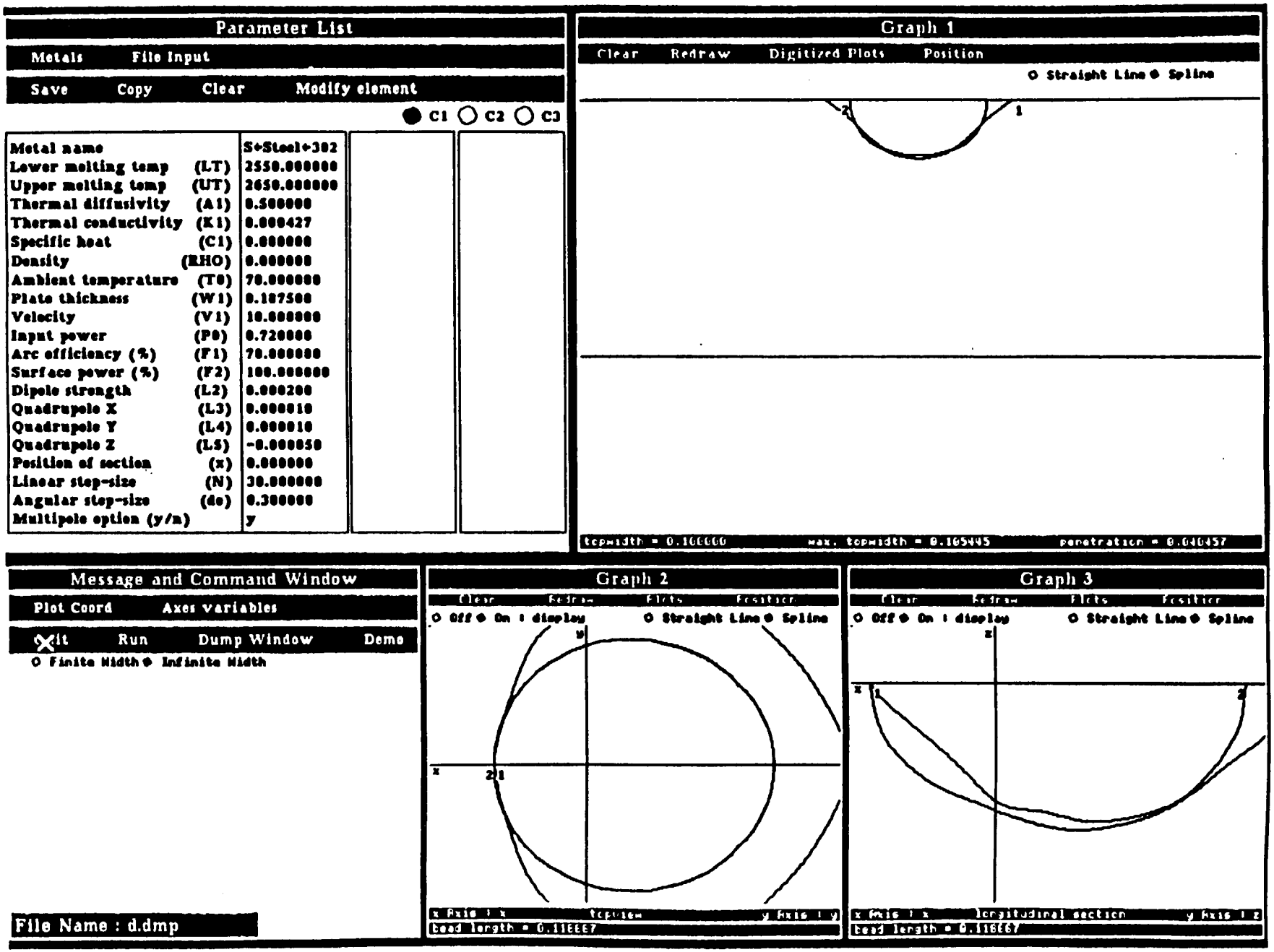


Figure 6. Actual (1) vs. simulated Nunes' model (2) weld for 100 Amperes.

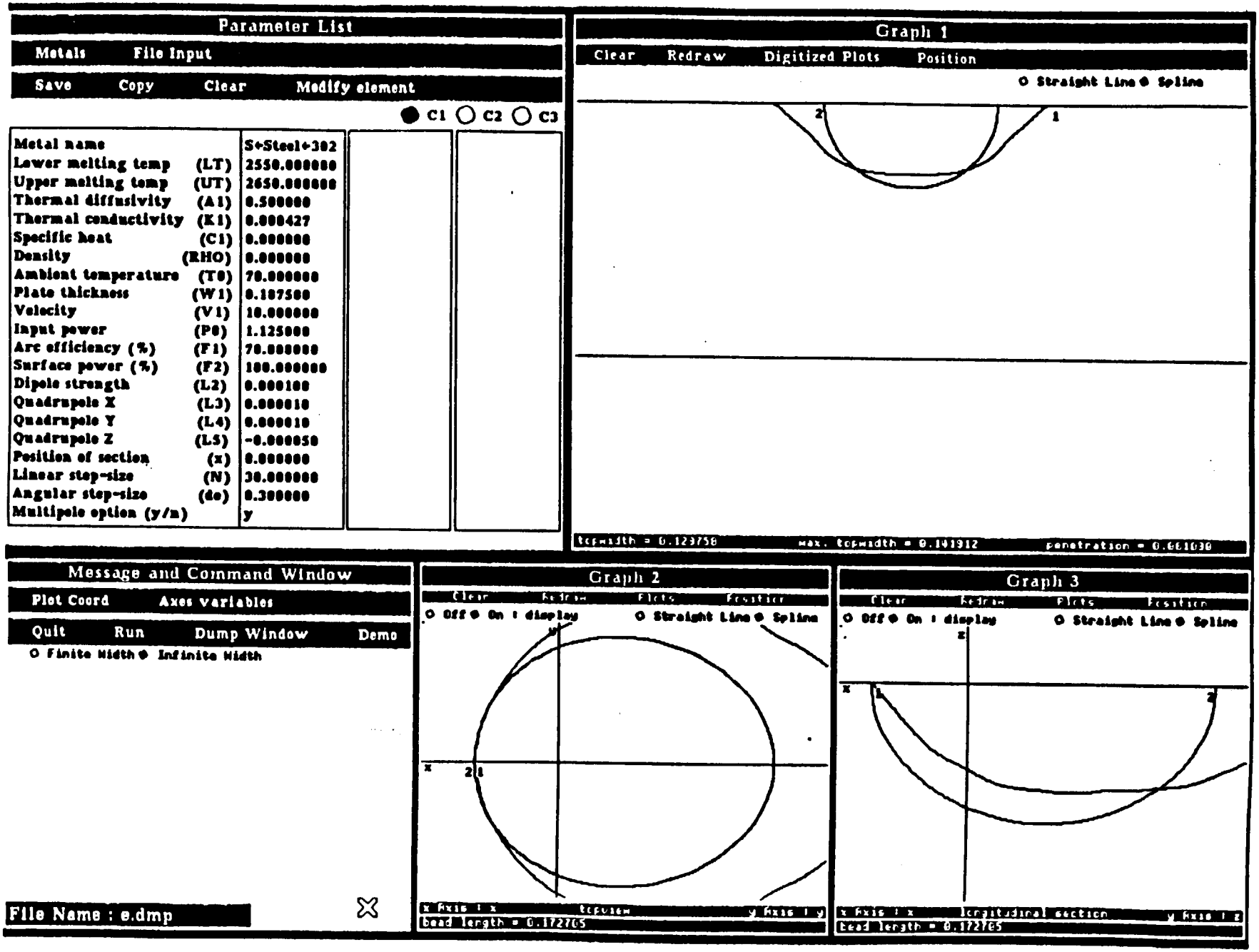


Figure 7. Actual (1) vs. simulated Nunes' model (2) weld for 150 Amperes.

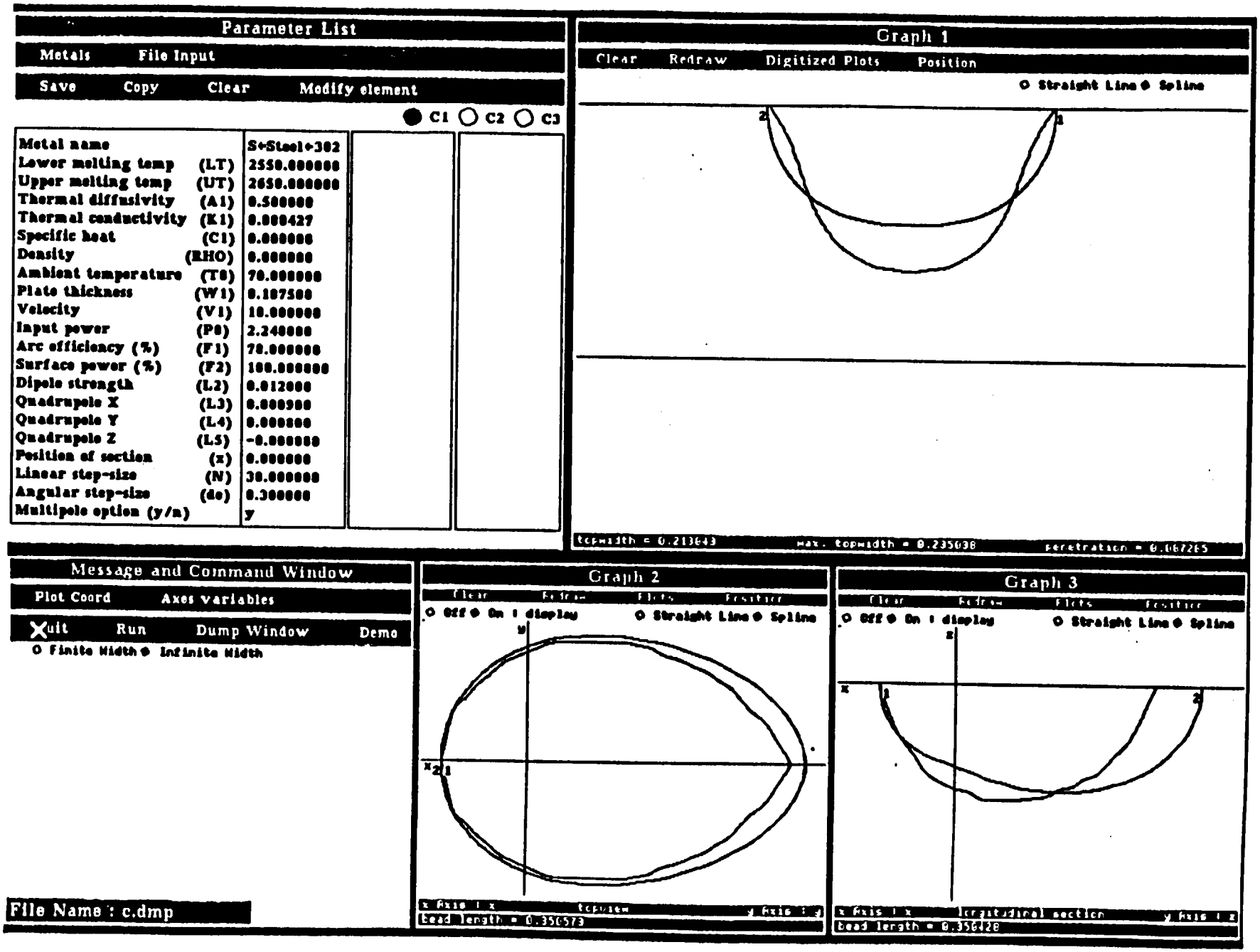


Figure 8. Actual (1) vs. simulated Nunes' model (2) weld for 200 Amperes.

The third, and present, version of the model is internally significantly different from previous versions. Previous implementations were to a large extent "translations" of the original Basic code, in the sense that the program structure was mostly the same with program branching making extensive use of "GOTO" statements, and variables used the same two-letter names as in the original Basic program. The third version was entirely rewritten and is more in line with conventional C programming style. This version is implemented in the Microsoft Windows system, which is being delivered to NASA. Examples of the modeling screens are shown in Figures 9 and 10. Both Nunes' model and Tsai's model use the same interface, and the user selects one model to be active at any given time. The modeling facility, just as other components of the Windows system, allows the user to work in either metric or English units. The user can switch between system of units at any given time, and the screen values are updated accordingly.

4.3 Tsai's Model

4.3.1 Underlying Theory Of Tsai's Model

The model developed by N. Tsai [24] is yet another extension of Rosenthal's model. Rather than expanding the heat source as dipoles and quadrupoles, as Nunes did, Tsai took the approach of representing the heat source as a disk, delivering maximum energy at the center and diminishing as the observation point is moved away from the arc center. Specifically, a cylindrical gaussian distribution of heat flow is assumed, as discussed earlier. Contrasted to the required knowledge of monopole, dipole, line, and quadrupole strengths of Nunes' model this representation requires specification of the total power and the standard deviation of the gaussian distribution for defining the weld puddle.

Tsai's model is closely related to Nunes' model, as both are improvements of the original Rosenthal model. In some respects this model is not as general as Nunes' model, due to its inability to model heat circulations in the molten weld puddle, as explained earlier. Additionally, Tsai's model does not account for a finite thickness of the workpiece. On the other hand the Gaussian representation of the heat distribution of the arc is very realistic and is supported by various experiments. Several researchers (Smartt et al. [25]) have verified that the arc heat in bead-on-plate experiments actually has an approximately gaussian heat distribution. The two input parameters of this model, the total power and the standard deviation of the Gaussian distribution, were found to yield fairly decoupled control of the bead geometry. The total power input appeared to govern the scale of the weld puddle while the arc deviation mainly affected the puddle width. One result of this is that the shape of the pool boundaries close to the weld surface is more accurately modeled. This was observed on several occasions where Tsai's model was able to

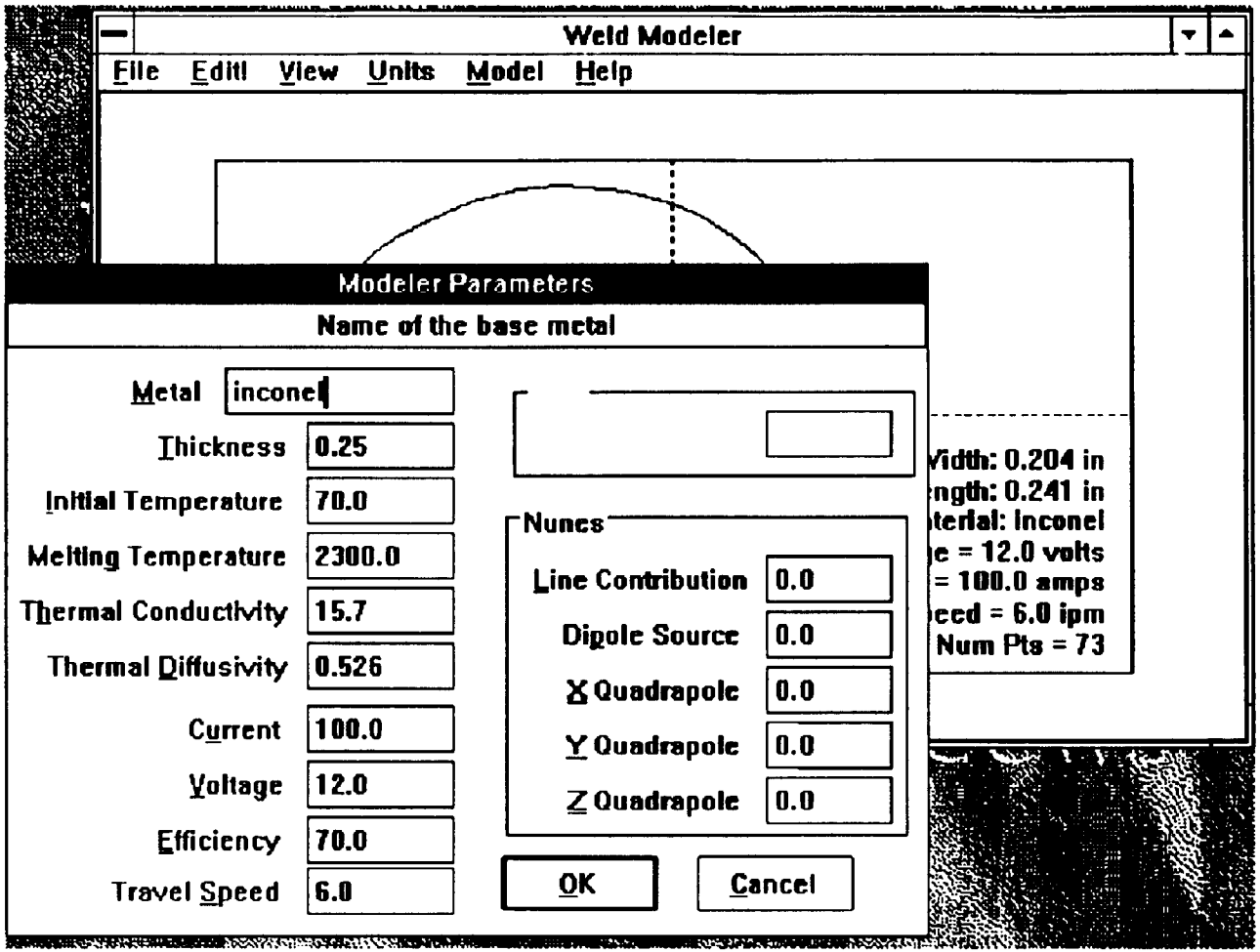


Figure 9. A sample model menu for parameter entry. Nunes' model is active. The section in the upper-right corner is reserved for Tsai's model, and is not printed out when Nunes' model is active.

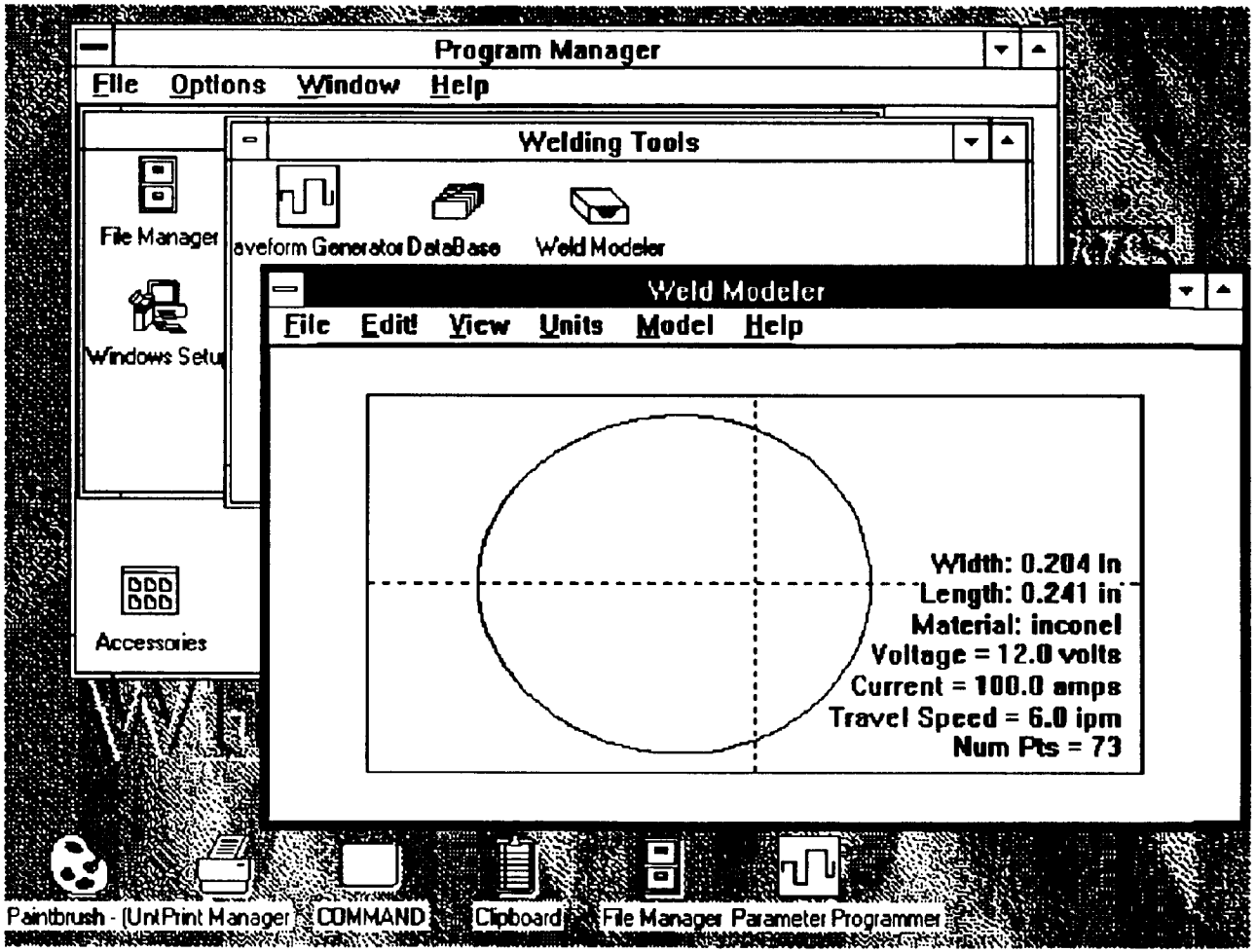


Figure 10. A sample of the model output screen, overlapping other Windows processes in the background.

model the "ears" or "wings" (molten pockets in the cross section at the plate surface) which Nunes' model is not capable of simulating.

4.3.2 Tsai's Model Implementation

The model is implemented in much the same manner as Nunes' model, and the user interface under Microsoft Windows is largely shared by both models. The only difference is that Nunes' line source, dipole, and quadrupole components are replaced by a measure of the arc width, the width "standard deviation".

4.4 Neural Network Model

The modeling technique presented here, using neural networks, is empirically based. The model is based on recorded welding data rather than the underlying heat flow physics. As illustrated at the end of the following section this approach has certain advantages as well as disadvantages as compared to more traditional models.

Recent successes in employing artificial neural network models for solving various computationally difficult problems have inspired revival of research in the area. Early work [26, 27] focused largely on mathematical modeling while more recent research has augmented theoretical analysis with computer simulations and implementation demonstrations. Numerous variants of pattern classifiers using neural networks have been studied [28, 29]. Introductory texts to the subject may be found in Rumelhart [30] and Lippmann [31].

As the name indicates, a neural network resembles, to a certain degree, biological nervous systems as we currently understand them. While most traditional computers rely on a single or few computational units to perform their tasks the neural network typically consists of a relatively large number of computational units, connected with an even larger number of communication links. The underlying principle aims to examine numerous hypotheses simultaneously and process data in a distributed fashion. In essence the neural network is a self-adaptive structure which incrementally alters its inner workings until it achieves the desired performance.

4.4.1 Neural Network Theory

A neural network and its adaptation procedure using back-propagation is best illustrated by an example. Figure 11 shows a small neural network consisting of 8 nodes, arranged in two hidden layers of 3 nodes each and one output layer of 2 nodes. Each node i in the first hidden layer produces a single numeric output which we denote as $x_i(1)$. Similarly the nodes of the second hidden layer are labelled $x_0(2)$

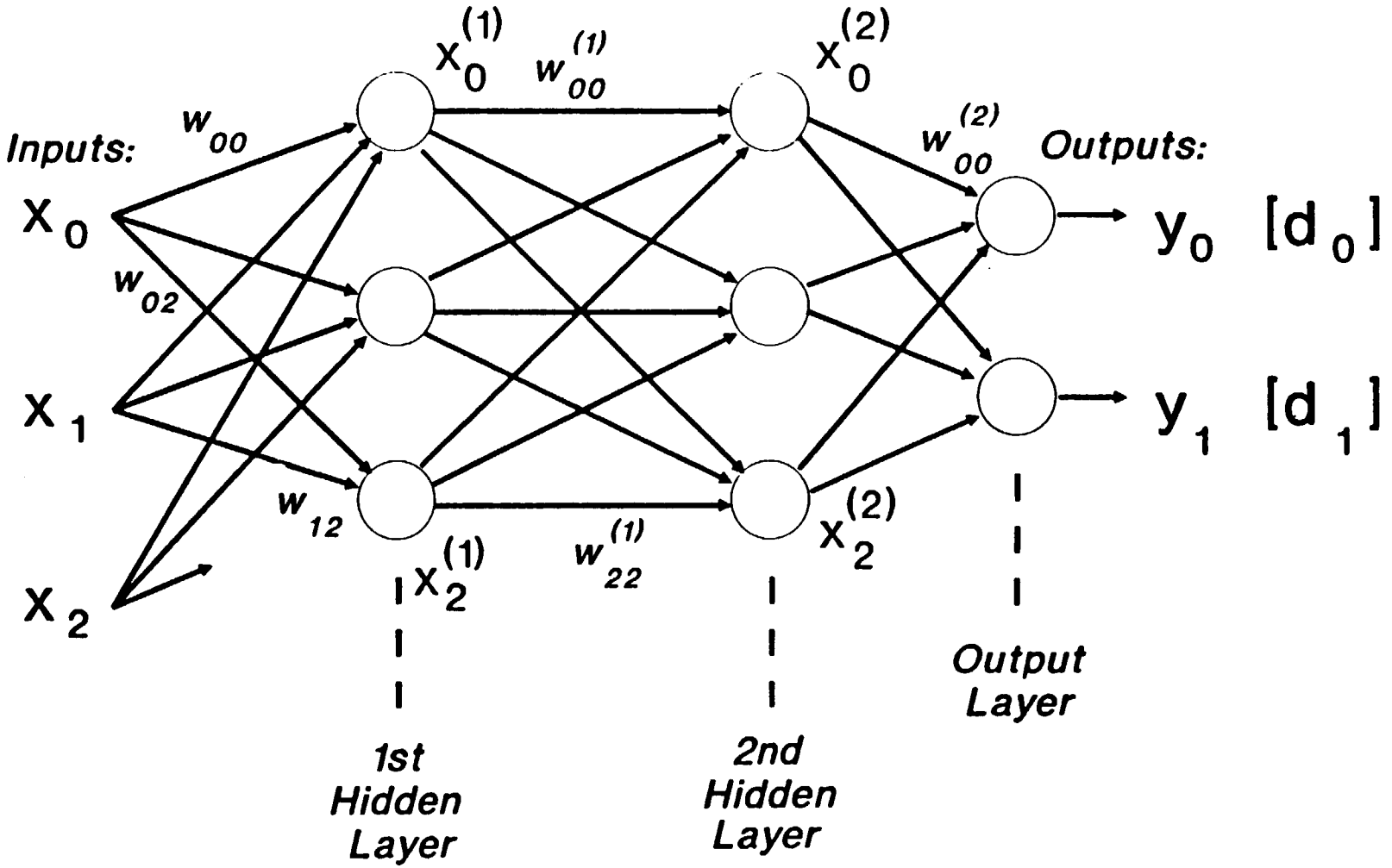


Figure 11. A 3-input, 2-output neural network, using 2 hidden layers of 3 nodes each.

through $x_2(2)$. The 3 inputs and 2 outputs of the network are x_0 through x_2 and y_0 through y_1 respectively. Each node accepts numeric data through a number of input links, each of which multiplies the input data with a weight factor. The weight factor associated with the link from $x_i(1)$ to the node producing $x_j(2)$ is annotated as $w_{ij}(1)$ and a similar convention holds for the links between other layers. Each node calculates its output by summing its weighted inputs and using the result s as the argument of a nonlinear function associated with the node. For our application this function is the same for all nodes:

$$f(s) = [1 + \exp[-(s-c)]]^{-1} \qquad (7)$$

where *s* is the sum of the node inputs and *c* is an internal offset value. Clearly the node output will be confined to the range $0<f(s)<1$. Because the limiting values, 0 and 1, will only be approached as s approaches + /- infinity all input and output data are scaled so that they are confined to a subinterval of [0...1]. A practical region for the data is chosen to be [0.1...0.9]. In this case each input or output parameter p is normalized as p_n before being applied to the neural network according to:

$$p_n = [(0.9-0.1)/(p_{max}-p_{min})](p-p_{min}) + 0.1 \qquad (8)$$

where p_{max} and p_{min} are the maximum and minimum values, respectively, of data parameter p. The network starts calculating its output values by passing the weighted inputs to the nodes in the first layer. The resulting node outputs of that layer are passed on, through a new set of weights, to the second layer, and so on until the nodes of the output layer compute the final outputs.

Before practical application, the network has to be *trained* to perform the mapping of the three input parameters to the two output parameters. This is done by repeatedly applying training data to its inputs, calculating the corresponding outputs by the network, comparing them to the desired outputs, and altering the internal parameters of the network for the next round. The training starts by assigning small random values to all weights (w_{ij}) and node offsets (c_j) in the network. The first three input data values are presented to the network which in turn calculates the two output values. Because the initial weights and node offsets are random these values will generally be quite different from the desired output values, d_0 and d_1. Therefore the differences between the desired and calculated outputs have to be utilized to dictate improved network values, tuning each weight and offset parameter through back propagation. The weights preceding each output node are updated according to

$$w_{ij}(t+1) = w_{ij}(t) + nd_jx_i^{(2)} \qquad (9)$$

where *n* is a *correction gain* and d_j is the *correction factor*

$$d_j = y_j (1-y_j) (d_j-y_j) \quad (10)$$

Clearly each weight will be increased if the calculated output from its node is less than the desired value, and vice versa. The correction factors used to update weights preceding hidden layer nodes are updated according to

$$d_j = x_j (1-x_j) \text{ sum}(d_k w_{jk}) \quad (11)$$

where the k applies to the node layer succeeding the one currently being updated. The offset parameter c of each node is treated as an additional weight factor and updated in the same manner.

The weights and offsets of the neural network are recalculated during the back propagation as outlined above. Then the network repeats calculation of output values based on the same input data, compares them to the desired output values, and readjusts the network parameters through yet another back propagation phase. This cycle is repeated until the calculated outputs have converged sufficiently close to the desired outputs or an iteration limit has been reached. Once the neural network has been tuned to the first set of input/output data, additional data sets can be used for further training in the same way. To ensure concurrent network adaptation to all sets of data the entire training process may be repeated until all data transformations are adequately modeled by the network. This requires, of course, that all the data sets were obtained from the same process and therefore the underlying input/output transformation is consistent.

As noted above, the training iteration process may be terminated either by a convergence limit or simply by limiting the total number of iterations. In the former case we use an error measure *e* defined as following:

$$e = \max_{k=1..K} \{\Sigma (d_{k,m}-y_{k,m})^2\} \quad (12)$$

where K is the number of input/output data sets used for training, M is the number of network output parameters in each data set, and $(d_{k,m}-y_{k,m})$ is the error in the network calculation of parameter m in data set k. The sum is taken for m=0 to m=M-1. The error measure, e, changes after each round of network weight adjustments. In the long run e decreases as the network is refined by training iterations. Using this indicator one can program the network to terminate the iterative tuning process as soon as e reaches some threshold value, e_0. Alternatively, a given network may not be able to reduce the error measure down to the specified e_0. In that case the iterations may be terminated by simply specifying a maximum number for them.

The *training mode*, as described above, is a precondition for actually applying the neural network in the *application mode*. In this mode entirely new input data is presented to the network which, in turn, predicts new outputs based on the transfer characteristics learned during the training. If this new data is obtained from the same local region of operation of the process as during the training phase, data from the input/output relations should be governed by the same underlying process and the neural network should perform adequately. The neural network is not updated in the application mode.

When compared to other modeling methodologies neural networks have certain drawbacks as well as advantages. For the drawbacks the most notable one is the lack of comprehension to the physics of the process. Relating the qualitative effects of the network structure or parameters to the process parameters is usually impossible. On the other hand most physical models resort to substantial simplifications of the process and therefore trade accuracy for comprehensability. The advantages of neural models include relative accuracy, as illustrated in the following section, and generality. If the training data for a neural network is general enough, spanning the entire ranges of process parameters, the resulting model will capture the complexities of the process, including nonlinearities and parameter cross couplings, over the same ranges. Model development is much simpler than for most other models. Instead of theoretical analysis and development for a new model the neural network tailors itself to the training data. The network can be refined at any time with the addition of new training data. Finally, the neural network can calculate its result relatively quickly, as the input data is only propagated once through the network in the application mode.

The above review of neural networks reveals how they can learn and implement the mapping between a number of input and output parameters. It should be noted that these parameter sets can be selected arbitrarily. For the purpose of welding control a specific neural network can be trained to derive IWPs from DWPs, while another network can yield DWPs from IWPs. The previous application is the weld pool modeling problem, where the neural network derives the geometrical attributes of the weld pool from the parameters of the welding equipment. The latter application can be directly used to determine the equipment parameters necessary to achieve a certain pool geometry. Practical implementation schemes for these two applications is discussed in the following sections.

Selecting the appropriate indirect weld parameters for a given weld geometry is a nontrivial task. Generally each parameter of the welding equipment affects a number of the geometrical attributes of the welding pool. Current, for instance, affects both weld width and penetration. The human welder or weld designer usually selects the indirect weld parameters based on previous experience, handbook recommendations, etc., and then fine tunes the selection with trial experiments. This approach can be complemented or replaced by a neural network, where the user

specifies the required weld geometry by the reference, or desired DWPs. Assuming that the network has been properly trained off-line with previous sets of actual weld parameters the network produces a set of IWPs which are used as weld equipment settings. If the network parameters include the ones which have principal roles in controlling the weld pool the welder can achieve the required DWPs with minimal experimentations or experience. Once this data is available the neural network can be refined by additional off-line training with the new data, and thus its characteristics are continuously updated as future welds are produced.

It is emphasized that this scheme applies to static parameter settings. Dynamic control of the welding process is left to the various controllers of the welding equipment, such as the automatic voltage controller (AVC), power source (usually constant current for GTAW), etc. Thus, it is the responsibility of these controllers to maintain the parameters, selected by the neural network, constant in presence of noise or process perturbations.

A few couple of observations and notes on the above applications are in order here. First, the neural networks are taught by examples, which eliminates the need for explicit mathematical modeling or similar manipulations. Secondly, the presently implemented neural networks do not provide dynamic models. The neural networks yield steady-state data or slowly varying setpoints, which in turn determine the operating points of the welding equipment.

As far as accuracy is concerned, static neural networks appear to yield quite satisfactory results. This is demonstrated in the following section.

4.4.2 Neural Network Implementation

To evaluate the accuracy of neural networks for weld modeling a number of simulations were carried out. Actual GTA welding data was used for this purpose. The data consisted of values for *voltage*, *current*, *electrode travel speed* and *wire feed speed*, and the corresponding *bead width*, *penetration*, *reinforcement height* and *bead cross sectional area*. A total of 42 such data sets (refer to Table 2) were used. 31 data sets were selected at random and used for training purposes while the remaining 11 data sets were presented to the trained networks as new application data for evaluation purposes. The 31 training sets have been listed together in Table 2 and marked with *[Tr]*. The remaining 11 data sets were used as new data for testing the networks after training.

Before training the networks for IWP-DWP mapping, decisions had to be made about fixed network parameters, such as the correction gain, n, and the error criterion, e_0. Preliminary simulation tests were carried out to examine how these parameters affected training convergence rate, and to determine feasible values for

the remaining tests. A value of 0.09 was found to be suitable for e_0; substantially larger values yielded inaccurate network mappings while smaller error criteria required considerably more training iterations. Similarly, preliminary tests showed a correction gain of about 0.1 to 0.2 to be suitable. The tradeoff there was slow training convergence at low gains against convergence instability at high correction gains.

Using $n=0.1$ and $e_0=0.09$, numerous network configurations for mapping the IWPs to DWPs were tested and compared. Both the required iteration numbers and mapping performances were examined for these networks. Because the error criterion was fixed, all networks were expected to perform the mapping with a comparable accuracy and therefore the first objective was to determine which configurations required the least number of training iterations. Networks consisting of 36, 48, 60, and 72 nodes were built, and for any total number of nodes various numbers of hidden layers were tested as well. For example, networks of 60 nodes were built with 36 nodes in a single hidden layer, 18 nodes in each of 2 layers, 12 nodes in each of 3 layers, and 9 nodes in each of 4 layers. The 31 sets of welding data, from Table 1, were used for training each of the networks. The number of training iterations required to reduce the convergence measure, e, down to 0.09 is illustrated in Figure 12. A training iteration is defined here as one round of network adaptation to all sets of training data. The Figure shows that, depending on network configuration, the number of training iterations ranges from just over 16,000 down to about 1,200. For each total number of hidden layer nodes there appears to be an optimum number of hidden layers which requires the least amount of training iterations. Although the training iterations take place off-line and are therefore irrelevant to on-line modeling, they can be fairly time consuming. CPU time for a typical case of 1500 training iterations was about 1 hour and 50 minutes on a Micro-VAX computer. Although the program was not optimized for speed and performed various auxiliary functions it is clear that training time optimization is advantageous for extensive experimentations.

Because the error criterion for all networks was the same, their performances were comparable. Table 3 illustrates the modeling performance of the 2-by-18 node network which serves here as an example of a typical case. The DWPs estimated by the network are listed with the corresponding error percentages. The standard deviation of the bead width, penetration, reinforcement height and cross sectional area errors are 8.01, 20.18, 3.97, and 7.10%, respectively. Figure 13 summarizes the performances of all networks for the four DWPs. The bars show the standard deviations of the modeling errors for the various networks using the 11 tested data sets. Most of the errors are on the order of 5-10%, with penetration errors typically in the 20% range.

The above network configurations were tested again with randomly selected data from the original training data set. Interestingly, the errors were generally not

PRECEDING PAGE BLANK NOT FILMED

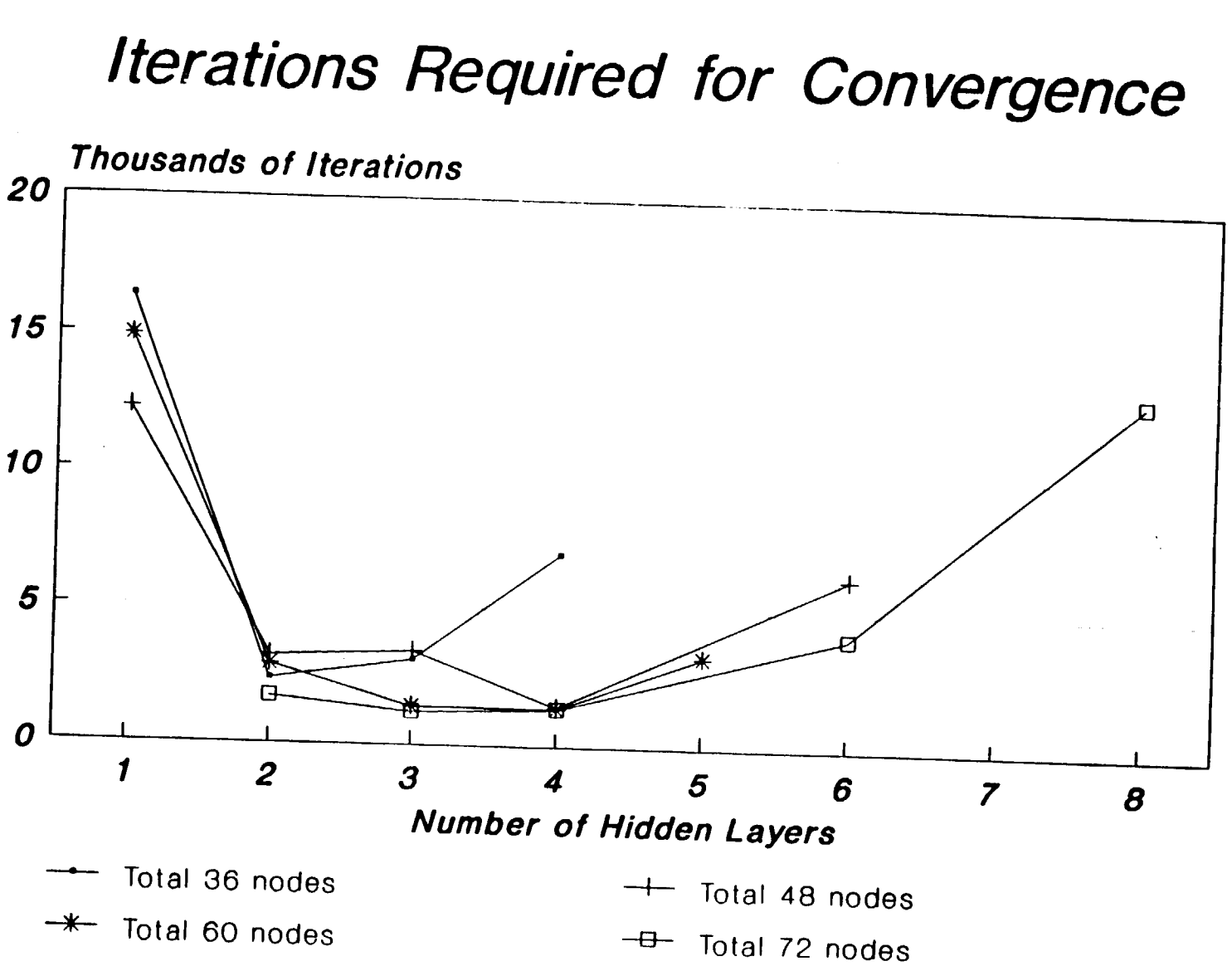


Figure 12. Number of training iterations required for convergences of various network configurations. Error threshold (e_0) is 0.09 and correction gain (n) is 0.1 for all networks.

Weld No.	Bead Width Est. [mm]	Bead Width Error [%]	Penetration Est. [mm]	Penetration Error [%]	Reinf. Height Est. [mm]	Reinf. Height Error [%]	Cross Sect. Area Est. [mm^2]	Cross Sect. Area Error [%]
32.	5.36	(0.19)	1.32	(-7.69)	1.75	(-7.41)	6.90	(-5.09)
33.	4.98	(-12.63)	1.65	(-4.62)	0.69	(0.00)	2.24	(-15.15)
34.	6.60	(8.91)	1.74	(-10.31)	0.96	(-1.03)	4.45	(5.20)
35.	7.65	(1.32)	2.94	(2.80)	0.91	(-4.21)	4.78	(-5.53)
36.	5.84	(1.04)	1.28	(-11.11)	1.96	(-7.98)	8.31	(-11.97)
37.	7.90	(1.28)	1.30	(-17.72)	0.41	(-6.82)	1.87	(1.08)
38.	8.09	(0.00)	1.22	(-32.22)	0.51	(-3.77)	2.	(-6.18)
39.	8.38	(1.33)	0.93	(6.90)	0.76	(1.33)	4.14	(3.50)
40.	8.37	(-16.96)	3.14	(45.37)	0.38	(-9.52)	2.20	(-13.73)
41.	10.00	(-2.34)	1.81	(2.84)	0.95	(-2.06)	6.77	(4.48)
42.	7.80	(-16.58)	2.59	(26.34)	0.31	(3.33)	1.57	(-9.77)

Table 3. Direct Weld Parameter Estimates and Errors from the 2-by-18 Neural Network. Refer to Table 2 for the Original Data.

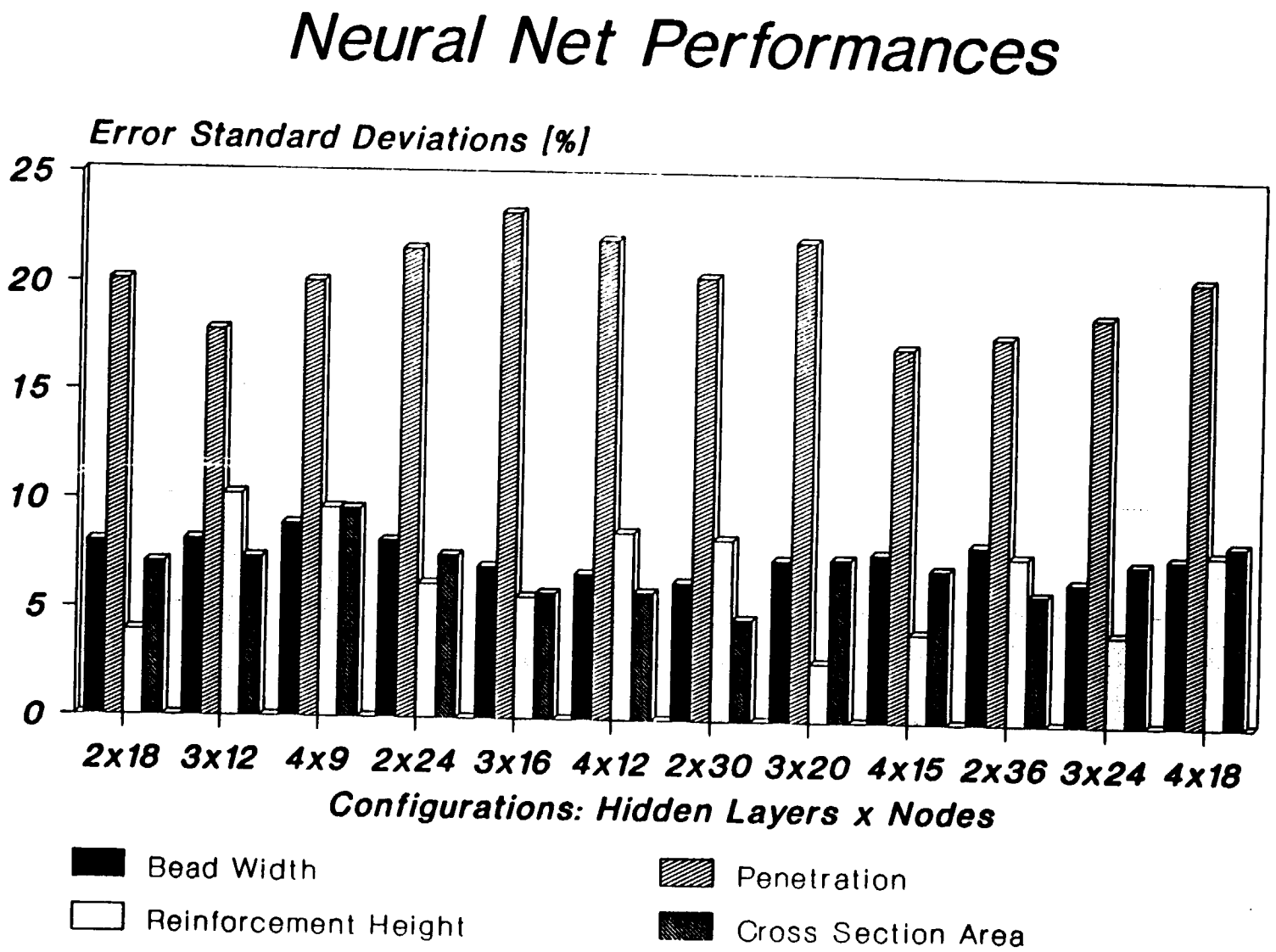


Figure 13. Standard deviations of mapping errors (in percentages) for various networks, shown for all four DWPs.

smaller than those obtained from the untrained data. Using the 2-by-18 network again for a comparison the standard deviations of the four DWPs are, in the same order as before, 2.86, 7.75, 7.13, and 6.72%. This observation underscores the fact that a network of a finite size can only model given data to a limited degree. In most practical cases network adaptation is terminated when a small residual mapping error is reached. Whether additional mappings, using new data, will be more accurate or not, depends on the nature of the data.

In summary, we have discussed the need for models of the welding processes and how neural networks can be employed for this purpose. Neural networks are very general in the sense that they can be constructed with empirical data rather than analytical formulations, which usually apply to specific cases. All fine tuning of the model to the data is built into the network structure, and additional data can be used for further refinements at any time. Although training of the neural network requires considerable computations, due to the iterative back-propagations, application of the network takes only a single forward pass and is therefore relatively fast.

4.5 Impulse Decanting For Model Calibration

Calibration and evaluation of a weld model requires accurate information on actual weld pool geometries. Metallographic analysis of the cross section of a weld bead has traditionally been used for this purpose and was used for significant portion of the data used in this research. A typical sample is shown in Figure 14. While the method yields fairly accurate information on the cross section normal to the travel direction, longitudinal and horizontal (pool top view) cross sections cannot be obtained by cutting the welded metal.

In order to obtain the maximum amount of information on pool geometry, the technique of "impulsive decanting" has been developed for use in determining instantaneous weld pool dimensions.

In the impulse decanting process, the pressure wave from the detonation of a charge of smokeless powder is directed at the molten weld pool and as a result the liquid metal is virtually instantaneously expelled from the base (unmelted) metal. The remaining crater is then a permanent record of the exact pool shape for the conditions being tested.

The apparatus used for this process is shown in Figure 15. The usual ceramic gas cup of the GTAW torch has been replaced with a brass cylinder which has been internally drilled to direct the gases from the cartridge toward the weld pool. A top view of the weld pool crater, after impulsive decanting, is shown in Figure 16.

A casting technique using low melting temperature metal was then developed which results in a positive (convex) impression of the pool crater. This al-

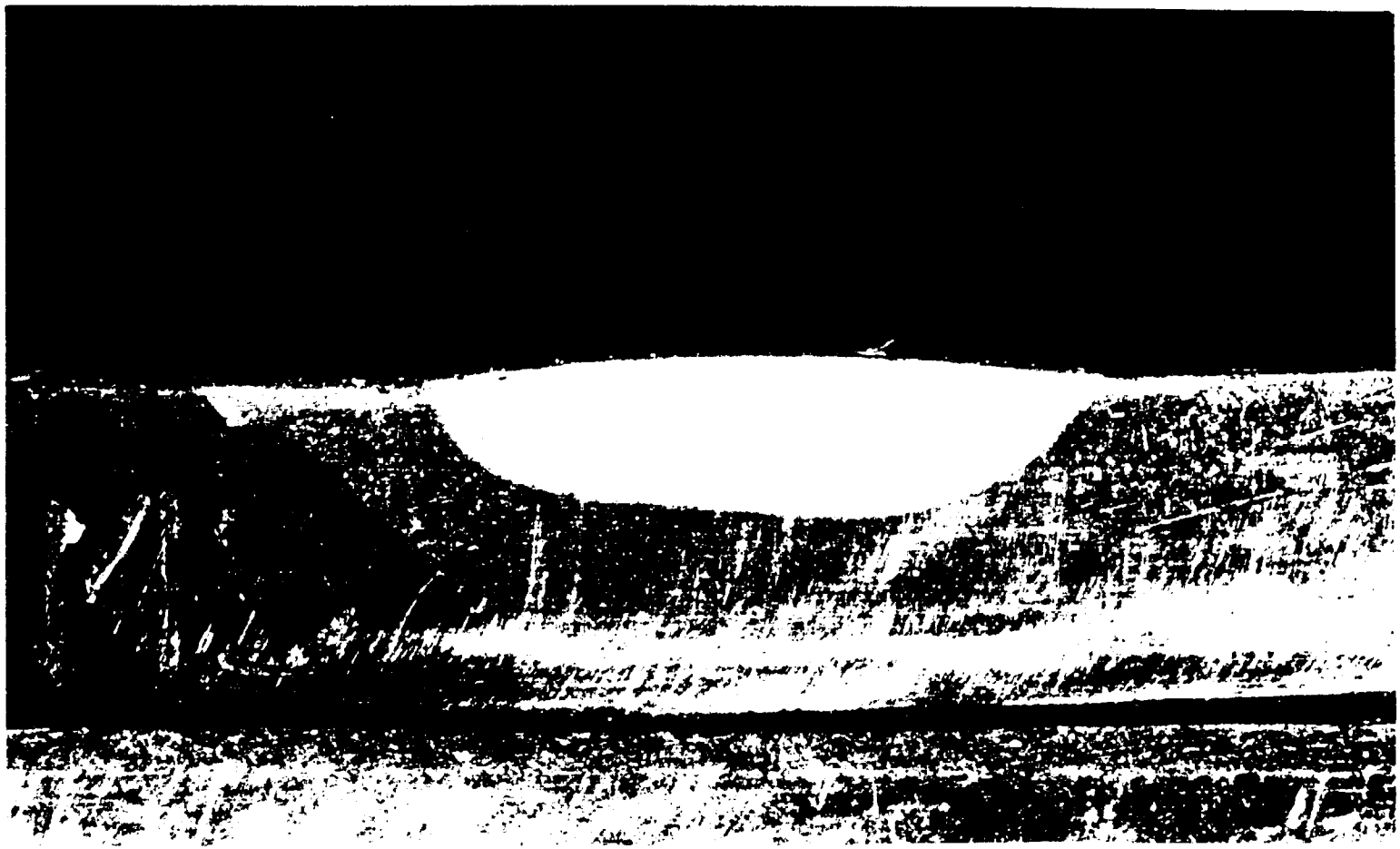


Figure 14. A metal sample cross section, used for metallographic analysis.

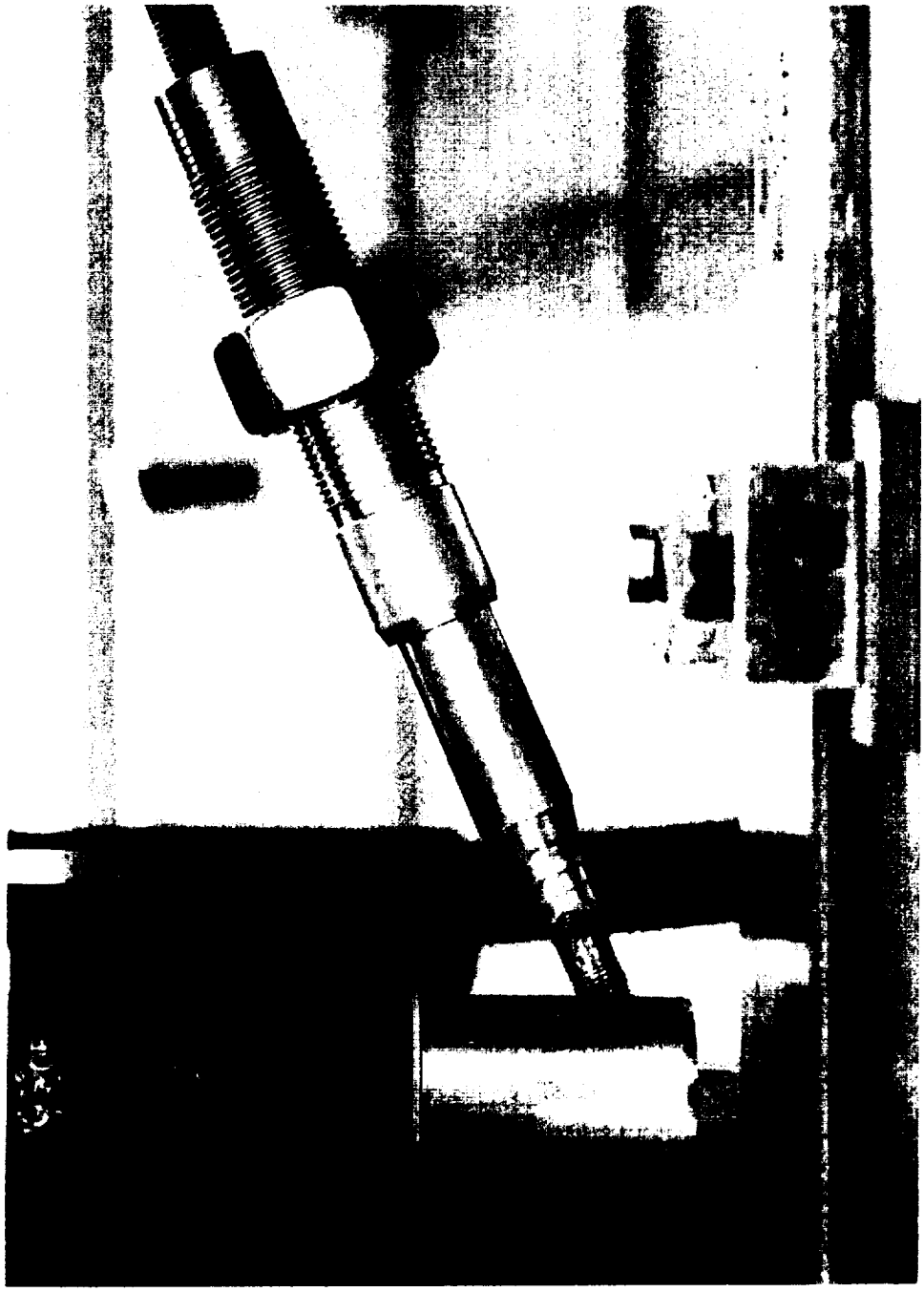


Figure 15. The impulse decanting welding torch.

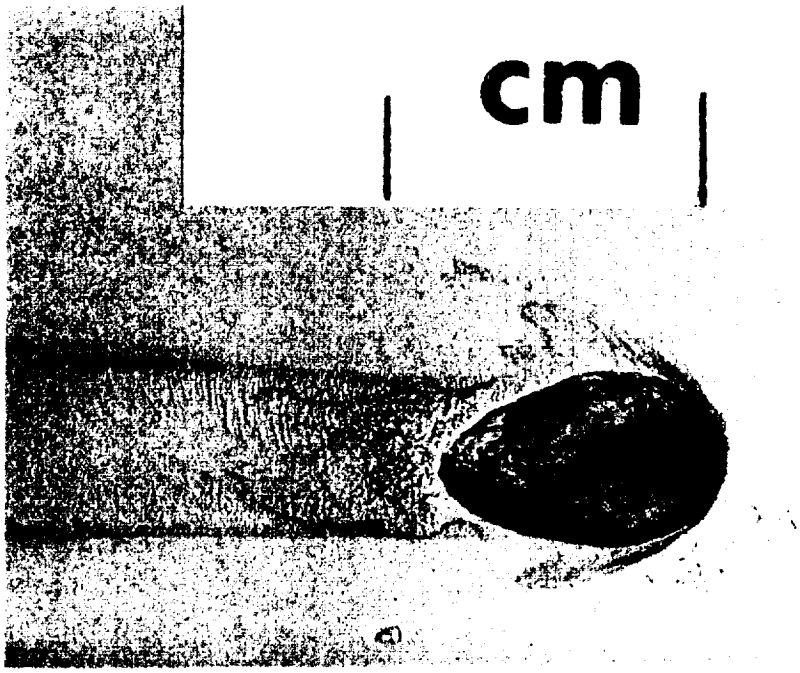


Figure 16. The top view of a decanted weld pool in Inconel-718.

lowed photographic recording of the weld pool dimensions for measurement and use in model calibration. In order to facilitate measurements of the lateral and longitudinal cross sections of the weld pool, photographic silhouettes of the principal planes (side view and end view) were taken and measured.

During development of the process, metallographic examination of the weld bead cross section adjacent to the crater showed that the expulsion efficiency was approximately 98%. After the apparatus was in operation, an earlier example of a similar experiment was found in the literature (Savage, et al., 1978 [32]). Their methodology and results were similar to the present study, with the exception that our use of a significantly larger powder charge eliminated the need for a separate timing circuit for extinguishing the arc, since the arc is immediately extinguished as the pressure wave passes by.

5. WELDING CONTROL

5.1 Introduction

One of the major goals of this research was to design and implement a welding controller, which provided the user with improved control capabilities beyond those of conventional welding controllers. A computer system has been assembled and programmed for this purpose and constitutes the deliverable items of this research.

The issue of welding control has been emphasized in two areas: (i) general control of the GTAW process welding parameters, and (ii) improved control of the automatic voltage control (AVC) system.

The research has demonstrated the feasibility of using weld models to determine the equipment parameters of the welding process. Further details on control for welding can be found in the recent paper by Cook, et al. [33]. Three models have been emphasized primarily for this work, i.e., Nunes' model, Tsai's model, and the neural network model. The neural network model is used in the delivered system to demonstrate weld parameter control. The physics-based models, implemented in the system, can be used to verify the viability of the neural network output parameters. Given a set of equipment parameters, determined by the neural network, the system has powerful capabilities for controlling all of the primary welding variables. Each variable can be programmed arbitrarily in terms of pulsing, sloping, or other types of variations. To facilitate such general programming, a graphical editor is provided where the user can graphically see the waveforms of the programmed current, voltage, arc length, travel speed, and wire feed rate.

The limitations of traditional AVC systems have been discussed at length with NASA and Rocketdyne personnel at MSFC, and improvement of the AVC system was a second area of control research and implementation. Through software implementation, rather the traditional analog design, the AVC can be made more robust and its capabilities can be greatly extended as compared with conventional systems.

5.2 Automatic Voltage Control

A substantial proportion of the SBIR work was directed towards analysis and improvements of the Automatic Voltage Control (AVC) system. The automatic voltage controller, common to most GTAW systems, is a typical feedback system. Figure 17 illustrates the main components of the AVC. With the GTAW process, the arc voltage is usually maintained at a specified level during welding. This voltage is sustained by feeding the actual arc voltage through a signal conditioner or filter, and comparing the resulting voltage with the reference voltage setting. The dif-

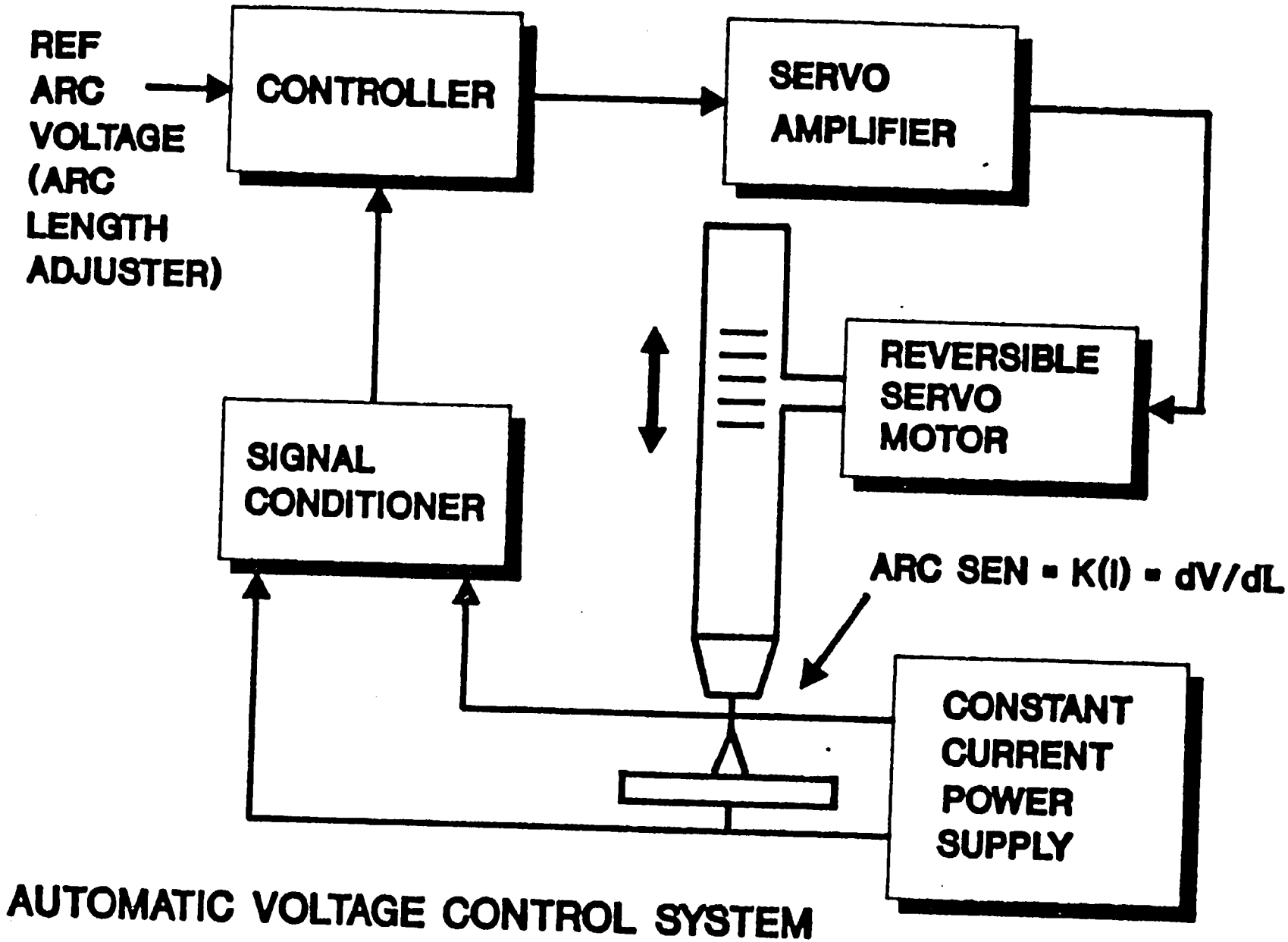


Figure 17. A typical automatic voltage control (AVC) feedback system.

ference (error signal) between the feedback and the reference voltage is amplified and used to adjust the servo mechanism which governs the Tip-to-Workpiece-Distance (TWD), or the arc gap length, as this distance is frequently referred to in this report. In this manner the arc voltage is kept constant at the setpoint level through continuous monitoring of the arc voltage and adjustment of the arc length.

This research on the automatic voltage control was primarily concerned with two aspects of the AVC: (i) improved control of the AVC mechanism, and (ii) capability for arc length control in addition to the more traditional voltage control. These two objectives are related in the sense that AVC improvements are necessary in the first place to achieve the arc length capabilities. The use of a digital control loop, implemented in software, permits more sophisticated control methodologies than traditional hard-wired analog controllers have offered.

Several visits were made to NASA-MSFC to learn about GTAW welding there and to discuss improvements for the AVC. General issues were discussed with Chip Jones, Art Nunes, Kirby Lawless, and Randall Shepard, all of NASA. Ben Coby, of Rocketdyne, demonstrated the GTA welding of Inconel-718 plates for the purpose of testing the strength of the joints. Information on the AVC systems used at MSFC was obtained from Ken Gangl and Doug Todd, also of Rocketdyne, who offered details of the AVC control aspects. Jack Weeks, of Rockwell, was particularly knowledgeable about AVC operation in general, and offered important insight into various problems that the welder encounters using the AVC. They all confirmed the undesirable behavior of the AVC during current upsloping and downsloping. To avoid problems during these periods the AVC has usually been turned off. However, because tail-out cracking is believed to be largely related to cooling rates and heat control in the solidifying pool, complete control of the total power input (voltage as well as current) is desirable at all times. The Rocketdyne and NASA personnel at MSFC were briefed on various AVC-related problems and possible solutions during the visits.

5.2.1 Welding Process Improvements

A primary goal of this project was to contribute to the enhancement of welds made with the GTAW process. Therefore, the digital AVC was developed for this primary goal more than to improve the operation of the mechanical welding equipment as such. The AVC developed during this research gives the operator improved control of the arc voltage and the arc length, which ultimately can be shown to result in improved welds, particularly during tailout of the arc.

Cracking problems, particularly in the tail-out regions where the current is sloped down prior to arc termination, were a primary reason for embarking upon the AVC research and improvement. It has long been known that the cooling rate

at the weld termination has a major effect on the susceptibility of the weld to cracking. By keeping tight control of heat input (i.e., both voltage and current) into the weld pool as the current is sloped down, rather than simply leaving the electrode at a fixed height and allowing the voltage to vary accordingly, the welder is given the capabilities to largely diminish or eliminate tail-out cracks.

The available welding literature was reviewed for research on weld cracking in GTAW and the general causes of cracking. Two papers were found on the welding of Inconel-718, which was of primary interest for this work. E.G. Thompson [34] of Rockwell Corp. noted that Inconel-718 is subject to heat cracking in the heat affected zone and that this cracking appears to result primarily from heat and be aggravated by large grain size of the alloy. Microcracking was determined to be directly related to the permanent impairment of the original elevated-temperature ductility, caused by exposure to a thermal cycle involving a degrading peak temperature. Specifically, the formation of low-temperature-melting intergranular films was noted as a result of exposure to temperatures ranging from 2100 to 2200 degrees F. The study by Gordine [35] on Inconel welding was more general and included the problems of poor penetration during welding, microfissuring in the heat affected zone, and poor impact and ductility properties of the weld fusion zone. Microfissuring in welded Inconel was found to be a function of (i) pre-weld solution treatment temperature, and (ii) heat input on welding. Examination of cracking close to welds made on Inconel-718 treated at 1700, 1900, and 2100 degrees F prior to welding, revealed evidence of cracking only on the 2100 degrees specimen. Gordine's ductility studies showed, furthermore, that impact strength (using the Charpy test) rose by a factor of 4 when the post-weld solution was increased from 1900 to 2000 degrees F. More directly related to heat control during welding, Gordine noted that "the susceptibility to microfissuring was greater under welding conditions of low heat input", without further elaborations. The post-treatment observations of Gordine are further confirmed by the findings of Mills [36] who studied the effect of heat treatment on the tensile and fracture toughness properties of Inconel weldments at various temperatures. He confirmed that a modified post-weld heat treatment at 2000 degrees F was superior to the conventional one at 1750 F, specified by ASTM A637. These studies manifest that pre-weld and post-weld treatments play significant roles in determining the overall strength of the welded metal.

Other research on cracking, in other materials than Inconel, was studied too. Savage and Lundin [37], in their discussion of the Varestraint test for hot cracking evaluation, noted a substantial increase of cracking in some steel samples when welding current was increased from 100 to 400 A, other factors being unchanged. Goodwin studied the effects of heat input and weld process on hot cracking in 0.010" thick 316 stainless steel. For this he compared GTAW, EBW (Electron Beam Welding), and LBW (Laser Beam Welding). Generally he found the welding parameters to have a significant effect on cracking response. Specifically, increasing

heat input turned out to increase cracking response. An interesting finding for the GTAW process was that cracking response was not simply a function of heat input, but also of the individual parameters determining the heat input. For example, relatively low current values at low travel speeds yielded much less cracking than higher current levels at higher welding speeds, although all cases resulted in the same net power or heat input to the weld. This was explained by the elongated pool shape at the higher travel speeds, affecting stress concentrations and alteration of solidification growth morphology at the trailing edge of the pool, and the fact that lower travel speeds allow a higher fraction of the heat input to be conducted away, resulting in a lower net heat input to the weld. Finally, although not directly related to this discussion, it was determined that both the EBW and the pulsed LBW processes provided overall improved cracking resistance over GTAW.

In summary, it can be concluded that heat input during welding, as well as pre- and post-weld treatments, determines to a large degree cracking and overall strength of the welded joint. Furthermore, the rapid temperature changes occurring in the vicinity of the endpoints of weld passes affect the solidification process and thermal stresses in these areas. A complete control of the heat input to the weld at all times, including current upsloping and downsloping, is therefore mandatory to aid suppressing of crack formation and other weld defects.

Graphically, Figure 18 illustrates the parameters and the process leading to tailout cracking. The equipment parameters selected by the welder during tailout (current, travel speed, etc.) determine the temperature field in the weld bead and the adjacent heat affected zone. The metallurgical processes that take place in the solidifying weld pool and in the heat affected zone determine, in turn, the microstructure of the bead and the adjacent base metal. The final microstructure is further shaped by post-weld treatment, if any is used. The microstructure of the metal finally determines its mechanical properties, and the amount and nature of defects, if any. Traditional AVC systems have not enabled welders to control the parameters which determine the integrity of the weld to the extent that is necessary to minimize tailout defects. When the traditional AVC is turned off as the rampdown is initiated, control is lost over both arc length and arc voltage. Together these parameters determine the total heat on the weld bead, as well as the input energy density. The time-varying heat input and energy density (which are generally decreasing during rampdown) determine the temperature gradients in the solidifying pool, which in turn govern the grain growth directions. Figure 19 illustrates the effects of different gradient patterns. The upper case shows a tear-drop shaped pool, where solidification takes place primarily from both sides of the pool. As a result, the grain directivity along the center of the weld, where the main solidification directions merge, is discontinuous. Cracks tend to form along the discontinuity on the center line of the weld. The lower part of the Figure, on the other hand, shows a temperature profile where grain growth directions in the solidifying pool are contin-

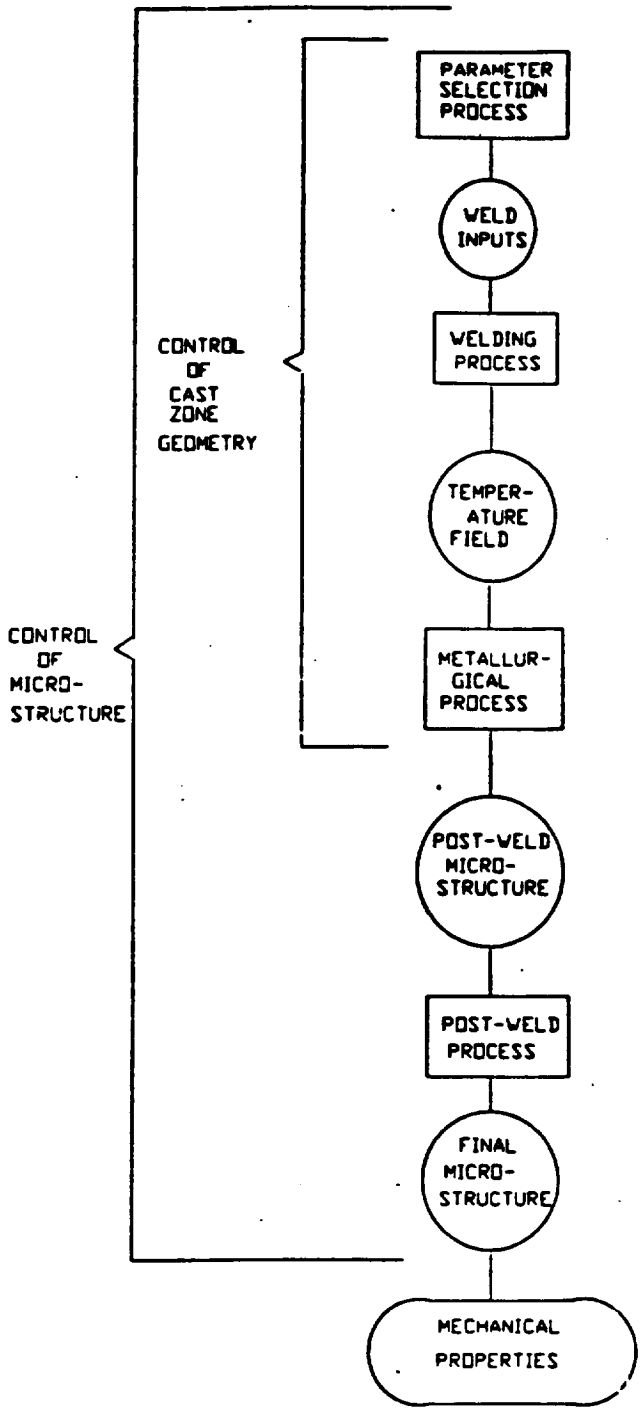
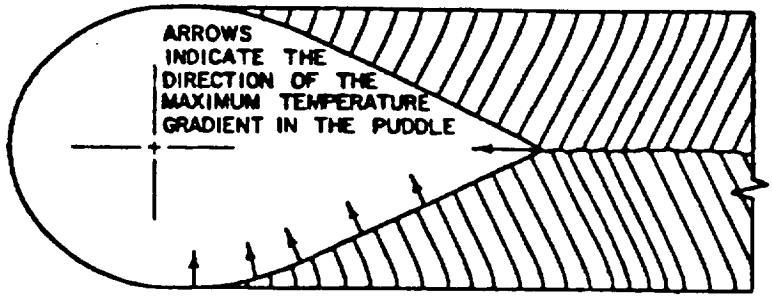
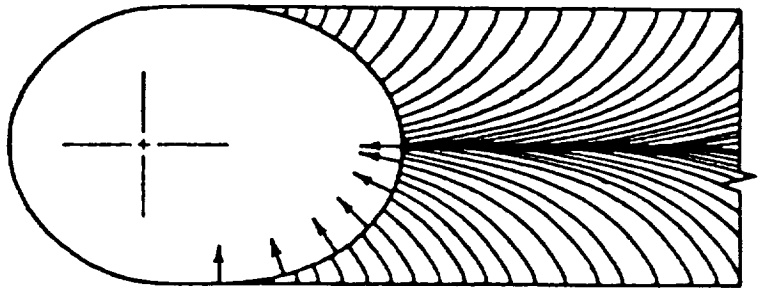


Figure 18. The processes and parameters which govern weld quality.



a. FOR TEAR DROP SHAPED WELD PUDDLE SHOWING ESSENTIALLY INVARIANT MACROSCOPIC GROWTH DIRECTION

WELDING DIRECTION



b. FOR ELLIPTICAL SHAPED WELD PUDDLE SHOWING CONTINUALLY CHANGING MACROSCOPIC GROWTH DIRECTION

Idealized schematic representation showing relationship between the macroscopic growth direction and the maximum temperature gradient in puddle

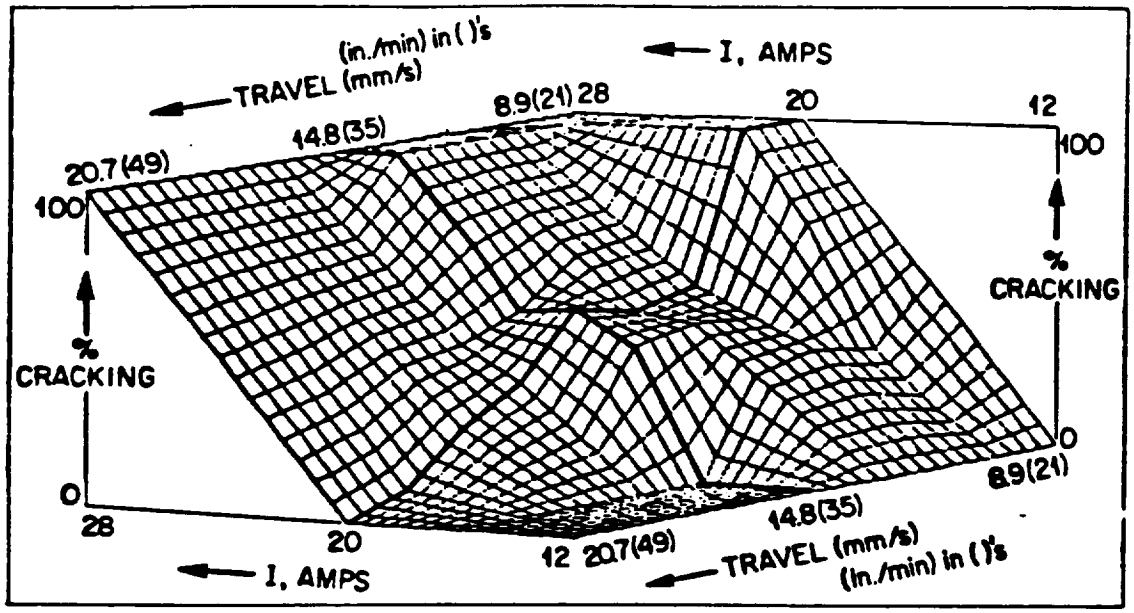
Figure 19. Effect of pool shape and resulting temperature gradient on grain orientation.

uous along the pool periphery. This is desirable and results in minimal cracking response. By reducing the arc length as the arc current is sloped down to zero, and thus decreasing the pool size at a desired rate, the idealized grain growth pattern can be attained on a real weld. This was done during the research period, and the welded pieces were demonstrated to NASA personnel. Use of the enhanced AVC is only one ingredient in the technique for reduced tailout cracking response.

An additional tool to reduce tailout cracking is varying, and specifically reducing, the torch travel speed as the weld is terminated. The travel speed is an additional parameter for controlling the temperature gradient of the solidifying pool, and promising welding results have been obtained varying the travel speed as the current and arc length are decreased. In general, control of travel speed is an important factor in controlling cracking tendency. This is demonstrated in the graph of Figure 20, where statistical cracking response is shown as a function of travel speed and current, for a given application. It should be noted that this Figure applies to cracking during steady-state welding, rather than during tailout. The developed welding control system facilitates arbitrary travel speed variations at any time during welding, including the tailout period.

Control of the arc length, to reduce the probabilities of cracking during tailout, is usually impossible using traditional AVC systems. Therefore, the AVC work centered on improving the overall stability, robustness, and general capabilities of the AVC. The deficiencies of AVC stability are briefly rehashed in the following. During representative welding conditions, when the arc currents and voltages are kept at nominal welding values, maintaining the welding voltage through the AVC is generally straightforward. Any "sluggishness" in the system can be adjusted to a required value through the adjustment of the tachometer feedback gain (a discussion of this and other control aspects will be given in the following subsection). It is during current upsloping and downsloping periods (typically during welding pass initiation and termination) that the AVC usually fails to perform as required. During these periods the torch position tends to become unstable; typically it oscillates or the torch runs the electrode into the workpiece surface. Because of these problems, the traditional AVC is usually disabled during the current upslope and downslope periods and the torch height maintained at fixed level.

In contrast to the traditional analog AVC, the digital AVC, developed here, can be fine tuned so that it can be maintained in operation while the arc current is tailed out. Furthermore, it can be programmed so that the welder can specify arc length, rather than arc voltage, for any segment of the weld. Use of the digital AVC can be particularly beneficial, as compared with traditional AVCs, for pipe welds and other circumferential welds where the weld is terminated where it was started. Such welds provide good examples for the need for an AVC that is maintained active during current downsloping. As the circumferential weld is terminated and the current is sloped down, the arc moves over surface of the beginning section of the



Cracking response to changes in current and/or travel speed

Figure 20. A graphical illustration of the effect that arc current and travel speed have on cracking susceptibility [47]

bead, which usually is elevated over the workpiece surface. If the AVC is turned off and the electrode tip is maintained stationary, the risk of running it into the existing bead is increased. Furthermore, the heat input and re-melting of the beginning section of the bead is uncontrolled and as a result the termination spot on the circumferential weld may be less than perfect.

5.2.2 The GTAW Arc

Thorough understanding of the arc voltage, and its dependence on other parameters of the system, is crucial for successful controller design. Therefore, numerous experiments were conducted to determine the relationships between the arc voltage, V_{arc}, the arc length, L, and the arc current, I. The results were used to determine which improvements were necessary to facilitate arc length control, and to improve the overall operation of the AVC so that it could be maintained active while the arc current was tailed out.

A number of experiments were carried out to determine: (i) V_{arc} as a function of L and I, and (ii) dV_{arc}/dL as a function of I and V.

A data sampling and processing system was developed to be used in conjunction with the adaptive AVC. It sampled arc voltages and currents in real-time and smoothed the sampled surfaces of voltages vs. current and arc length. The experiments were done with bead-on-plate welds on Inconel-718. A 3/32 (0.093) in. diameter, 2% Thoriated Tungsten electrode was used, with a 36 degrees included (total) tip angle. 100% Argon at 25 cfh was used for shielding. An arc was established at the lowest sustainable current, 5 A, and the distance from the electrode tip to the workpiece was reduced to the lowest attainable level without extinguishing the arc, which was 0.005 in. The voltage was measured between the collet holding the tungsten electrode and the Inconel block. Then the arc gap distance was increased to 0.010 in, maintaining the same current, and the corresponding voltage recorded. This was repeated for a number of successively increasing arc lengths. Next the current was increased to 10 A. The electrode was moved down to 0.010 in. (the arc could not be sustained at 0.005 in.), where the voltage was recorded and then again for a number of larger arc lengths. This sequence was repeated for higher current values as well.

The arc voltage as a function of arc length was determined experimentally and it is illustrated in Figure 21 (one curve for each current level). Second degree polynomials were used to interpolate the measured data points for each current level. This data was used to obtain the dV/dL curves as functions of current, for fixed voltage levels, as shown in Figure 22. For each selected voltage level the slopes of the various constant-current interpolation curves were computed, and the resulting values were used to set the loop gain of the digital AVC.

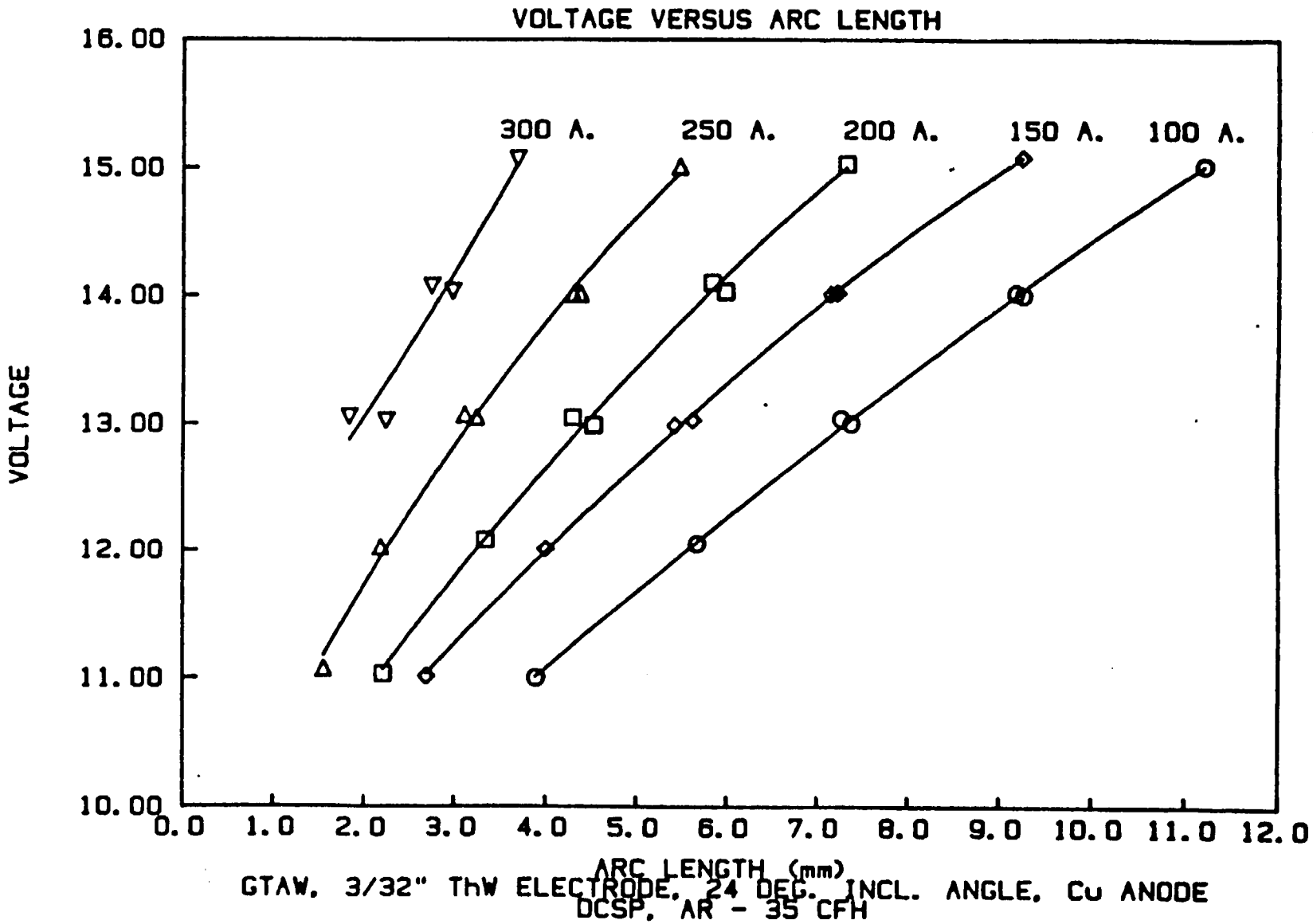


Figure 21. Arc voltage shown as a function of arc length, for various current levels. Shielding gas: Argon.

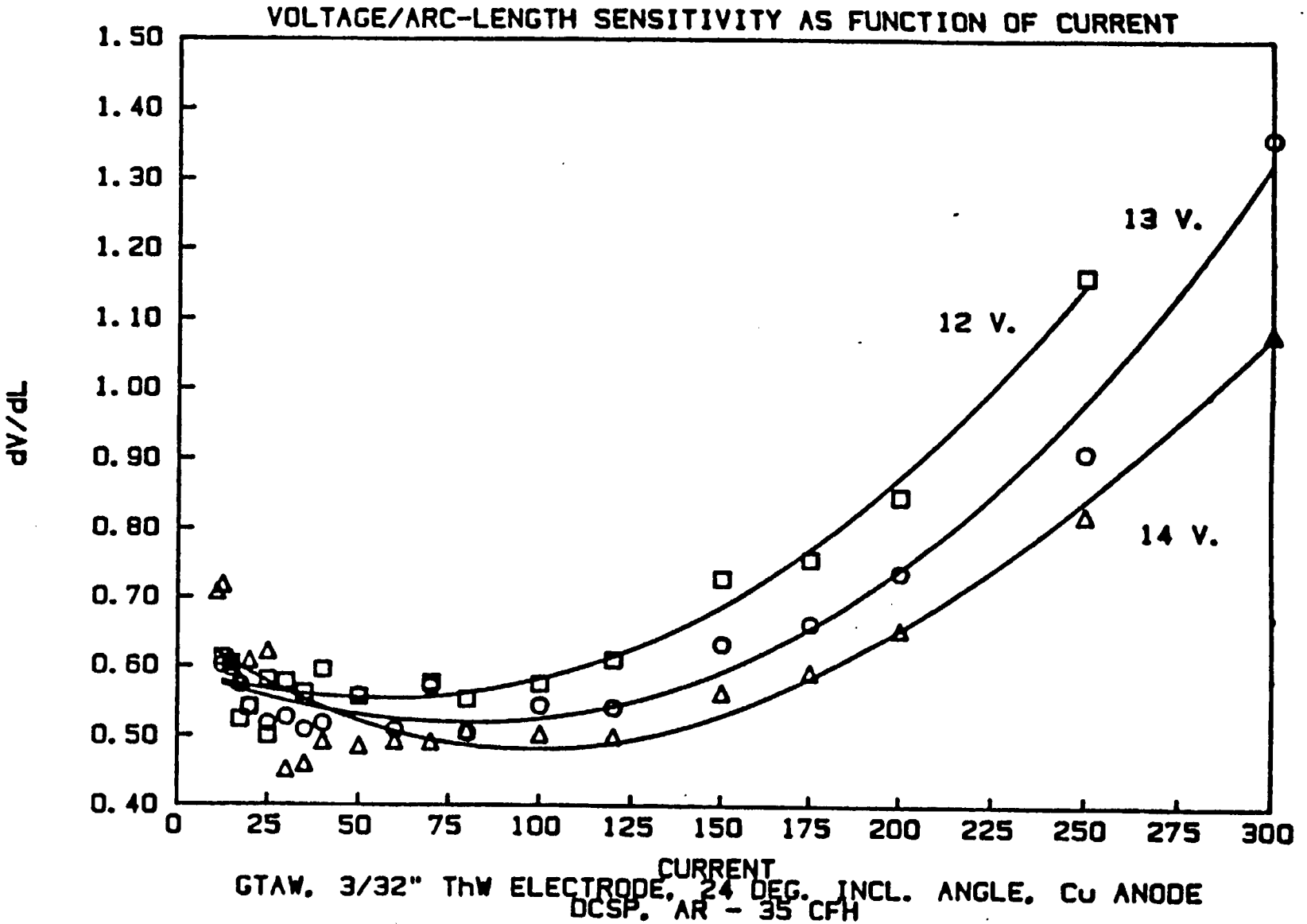


Figure 22. Arc voltage-to-length sensitivity (dV/dL, or K_v) compared for arc voltage of 12, 13, and 14 Volts. Shielding gas: Argon (35cfh).

A V_{arc} vs. L and I graph (Figure 23) illustrates a part of the AVC problem during current upsloping and downsloping. Assume that welding takes place at 100 A and 10 V. The corresponding arc length is about 0.200 in. Now, envision the current being sloped down while the AVC is left intact. The AVC forces the operating point on the graph to move along the horizontal 10 V level only. In this instance the operating point moves first to the right, as illustrated by the 10 V / 70 A point. This means that the AVC must temporarily increase the arc length to about 0.250" to maintain constant voltage. As the current is sloped further down the operating point moves back to the left and at 20 A the arc length has been reduced to about 0.020". Further current reduction eventually results in the electrode tip touching the workpiece surface. Data for shorter arc gap distances than indicated on the graph could not be obtained in the experiments, largely because of the dynamics of the pool surface which occasionally reached the electrode tip. This experiment, however, supports the general experience with most AVC systems which frequently run the electrode into the workpiece when the current is sloped down to extinguish the arc. Similar arguments apply to upsloping of the current as well. In both cases it is clear that the AVC does not serve well as a constant-arc length controller when the current is varied over a wide range, as is the case during upslope/downslope or tapering during the weld period to account for varying conditions of part geometry or heat buildup.

As pointed out earlier, the arc length can not be controlled through the arc voltage alone. Figure 24 illustrates the general features of arc voltage with respect to arc length and current, without specific values. The surface shown in this Figure is obtained from curve-fitting to real welding data, and the grid segments forming the surface are indexed in increments of 10 Amperes and 0.010 in. The equation describing this surface is:

$$V = 0.013\,I + 5.2 + 185/I + 10^{-3}\,(20.92 - 0.25I + 0.0017I^2)\,L \qquad (13)$$

The Figure shows that the arc voltage generally increases as the arc length is increased, as well known. Additionally, however, the arc voltage, at a fixed arc length, varies substantially as the arc current is varied, particularly as the current becomes small. The implication of this is essentially that a traditional AVC, which does not account for the voltage-current relationship, fails when the current varies substantially, such as during current rampdown or pulsing. This will be explained in the following paragraphs. Finally, the slope of the surface along the length-axis (dV/dL) is indicated in the Figure. This slope, although almost constant at any given arc length, varies substantially with current. This slope is essentially the arc sensitivity, which plays a major role in setting the dynamic response of the AVC.

Figure 25 shows the same voltage characteristic as Figure 24 does, but with a fixed, selected voltage, highlighted. A traditional AVC which is set to maintain this voltage at all times will track this trajectory on the surface, by adjusting the arc

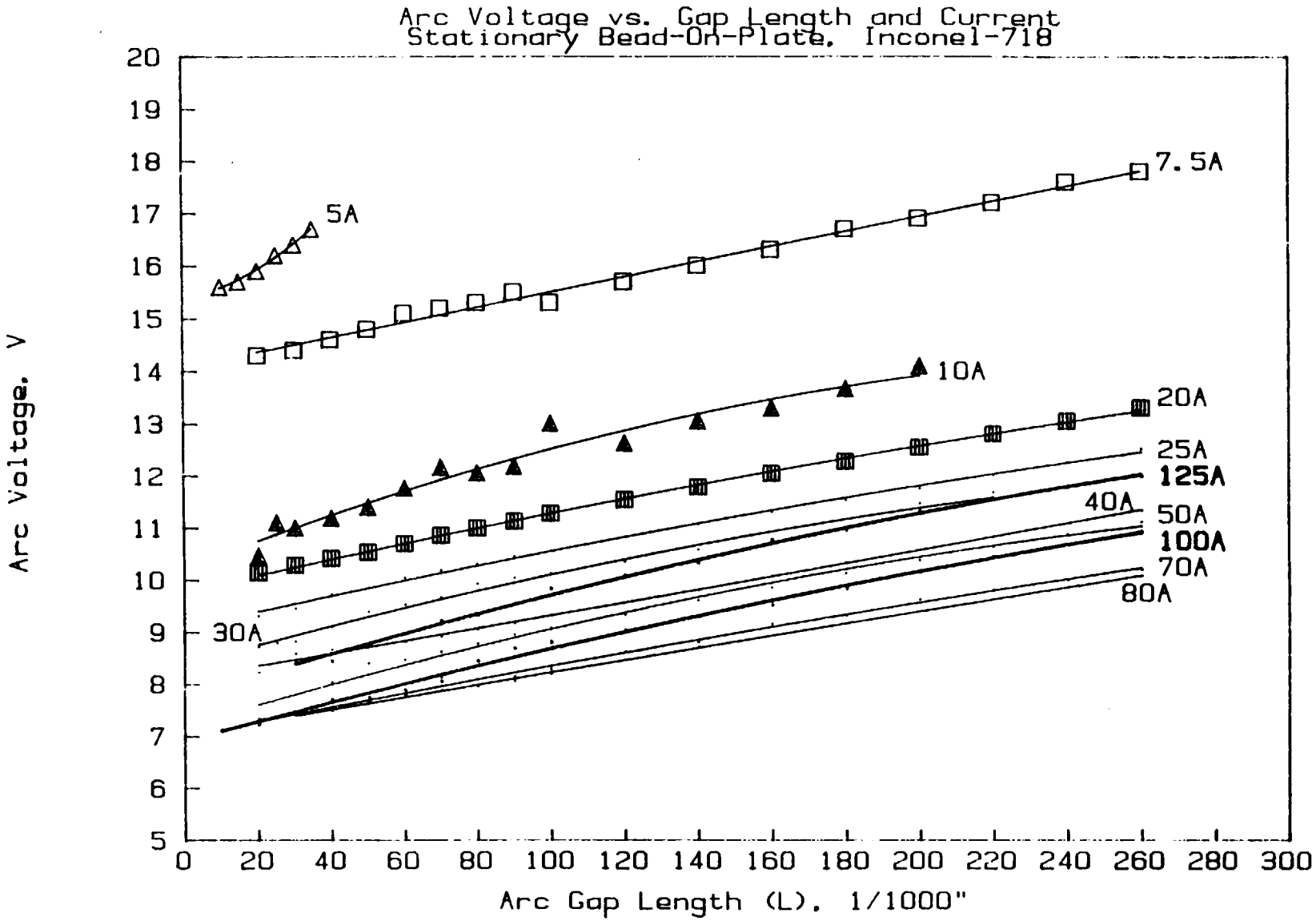


Figure 23. Arc voltage vs. arc length and current. This graph can be used to visualize the behavior of a conventional AVC during current tailout.

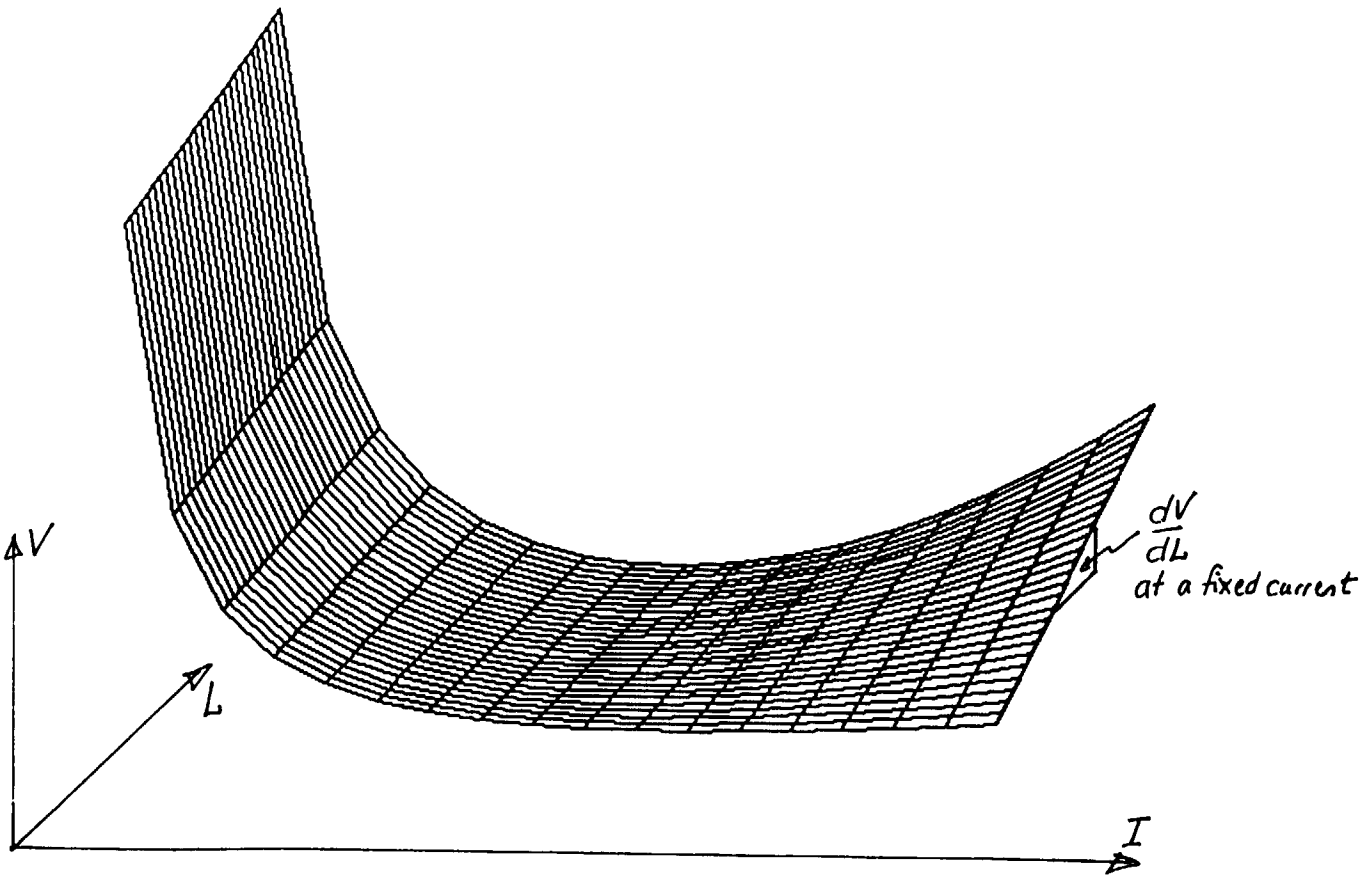


Figure 24. A three-dimensional view of arc voltage as a function of arc length and current.

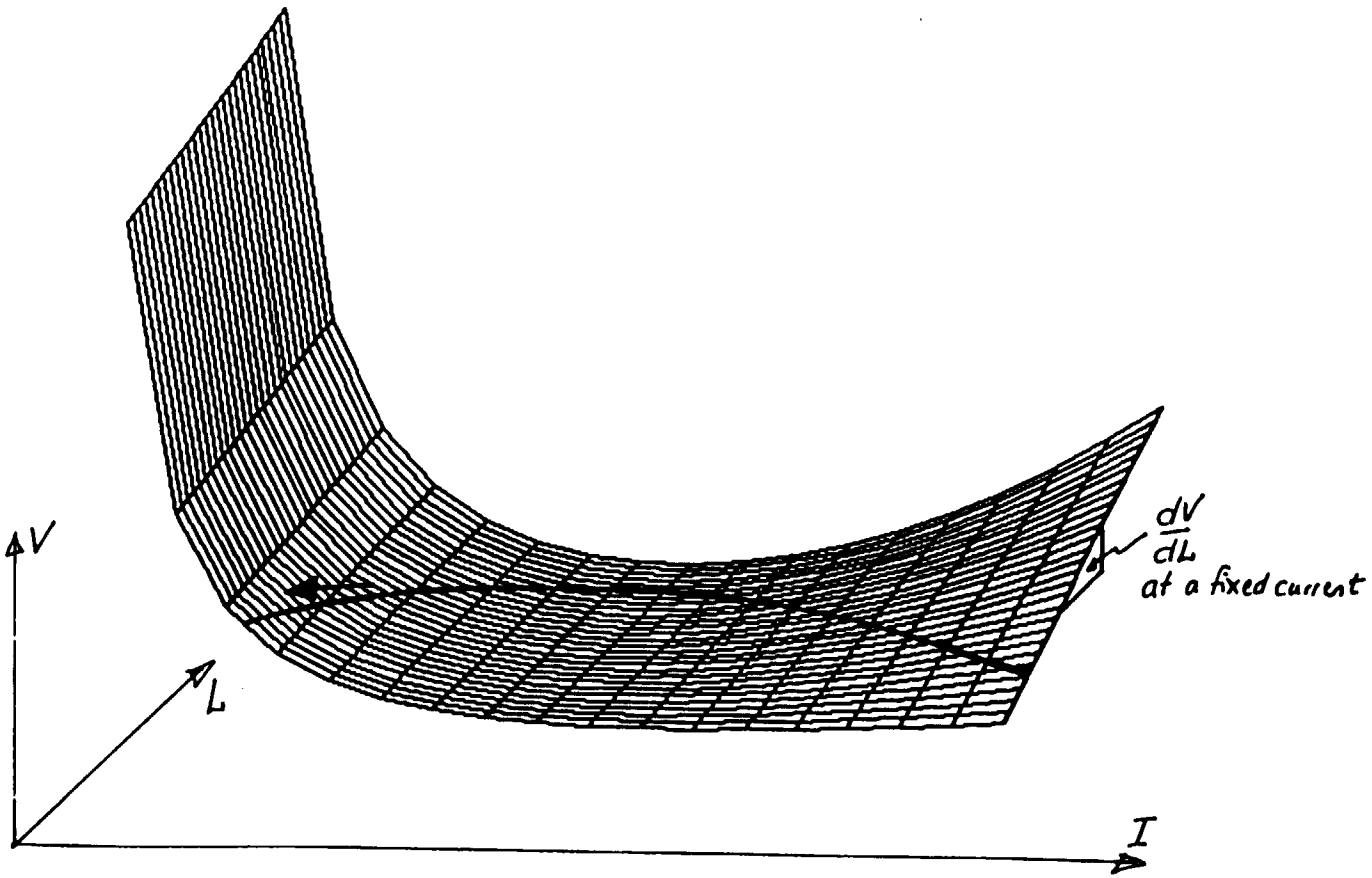


Figure 25. A three-dimensional view of arc voltage as a function of arc length and current. A selected, fixed voltage level is indicated by a highlighted curve.

length, as the current is varied. Although the arc length remains fairly constant at currents in the middle range, it decreases rapidly as the current is decreased. Notably, the arc length approaches zero (i.e., the electrode tip runs into the workpiece surface) when the current reaches a certain low value. To avoid this, traditional AVCs are usually disabled during current rampdown periods, such as for arc termination.

The above discussion on arc voltage as a function of current and length can be repeated in terms of arc length as a function of current and voltage. Figure 26, which shows arc length (vertical axis) as a function of voltage and current (horizontal axes), is obtained by rotating the surface of Figure 24 so that the voltage, current, and length axes are reoriented. This Figure shows how the arc length has to be increased if the arc voltage is to be increased, for a given constant current. Again, the length dependency on current is shown to be particularly pronounced at low currents. Note that the curve fitting procedure used to obtain this graph extrapolates the arc length surface into "negative" arc lengths in the region of very low currents and low voltages. Effectively, the voltage-current coordinates which are shown to yield negative arc lengths on this graph represent those voltage-current pairs which can not be attained by the arc at all. In other words, the electrode tip is driven down into the workpiece if the AVC and the current controller attempt to achieve these current-voltage combinations.

Figure 27 illustrates specifically how the arc length of a traditional AVC varies as the arc current is ramped linearly from a fixed operating value down to zero. The three-dimensional surface describing arc length is used again to illustrate this, but here a trajectory of the operating point has been sketched as well. The traditional AVC must vary the arc length as shown by the highlighted trajectory as the current is ramped down, if the constant voltage is to be maintained. Again, this leads to the electrode running into the weld pool as the current approaches the lower limit and the electrode tip is driven towards the workpiece in order to keep the voltage constant. The implication of this is that arc voltage and arc length can not simultaneously be maintained constant while the current is reduced. If one is held fixed, the other has to vary, to accommodate for the physics of the arc which determine the arc voltage-current-length relationships.

Because of the arc length relations to both current and voltage, the welder has to determine which of these two is to be directly controlled while the current changes. The term "direct control" may refer to keeping the variable at a fixed value or varying it in a well defined manner, e.g., ramping the voltage or arc length at a constant rate. There are basically two approaches to accomplishing the control: (1) the voltage is controlled directly while the arc length is allowed to vary as necessary, or (2) the arc length is controlled directly while the arc voltage is allowed to vary. Which of these options is selected should be based on which approach results in the more favorable microstructure of the molten metal in the pool, e.g., which op-

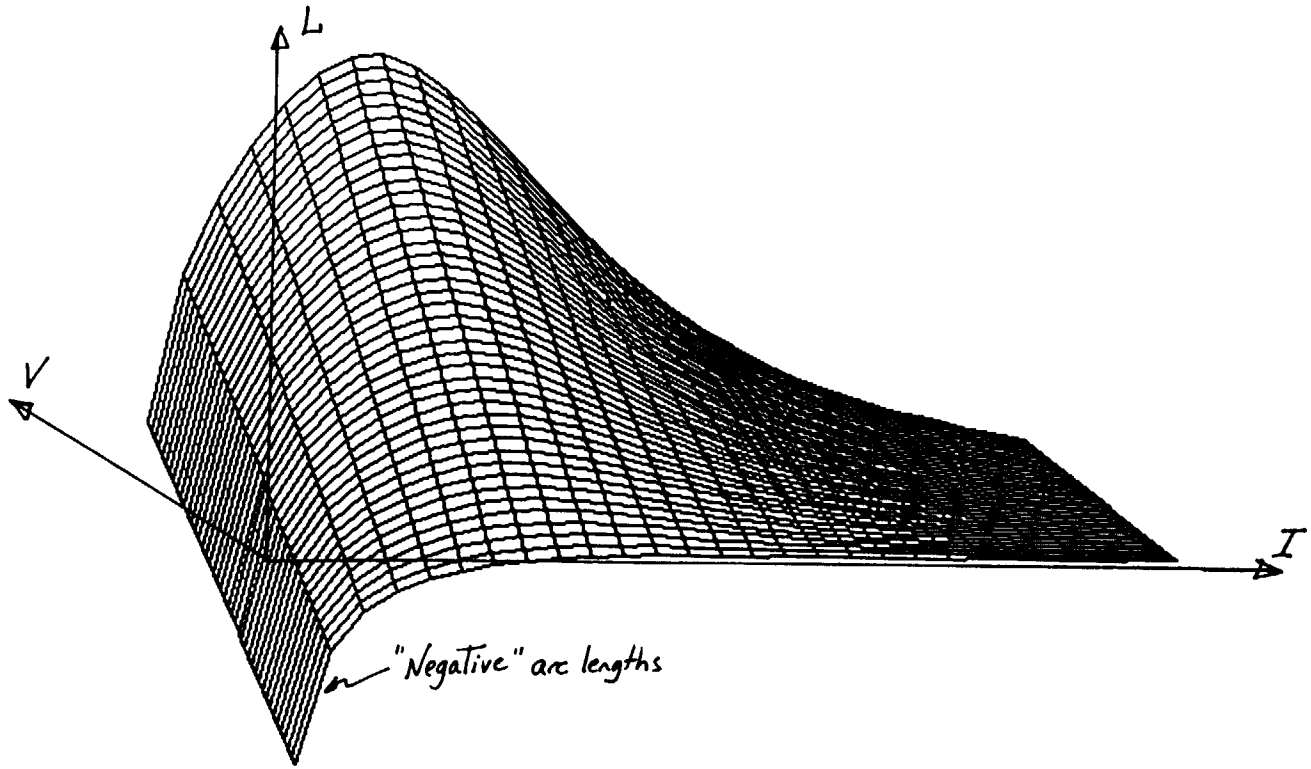


Figure 26. A three-dimensional view of arc length as a function of arc voltage and current.

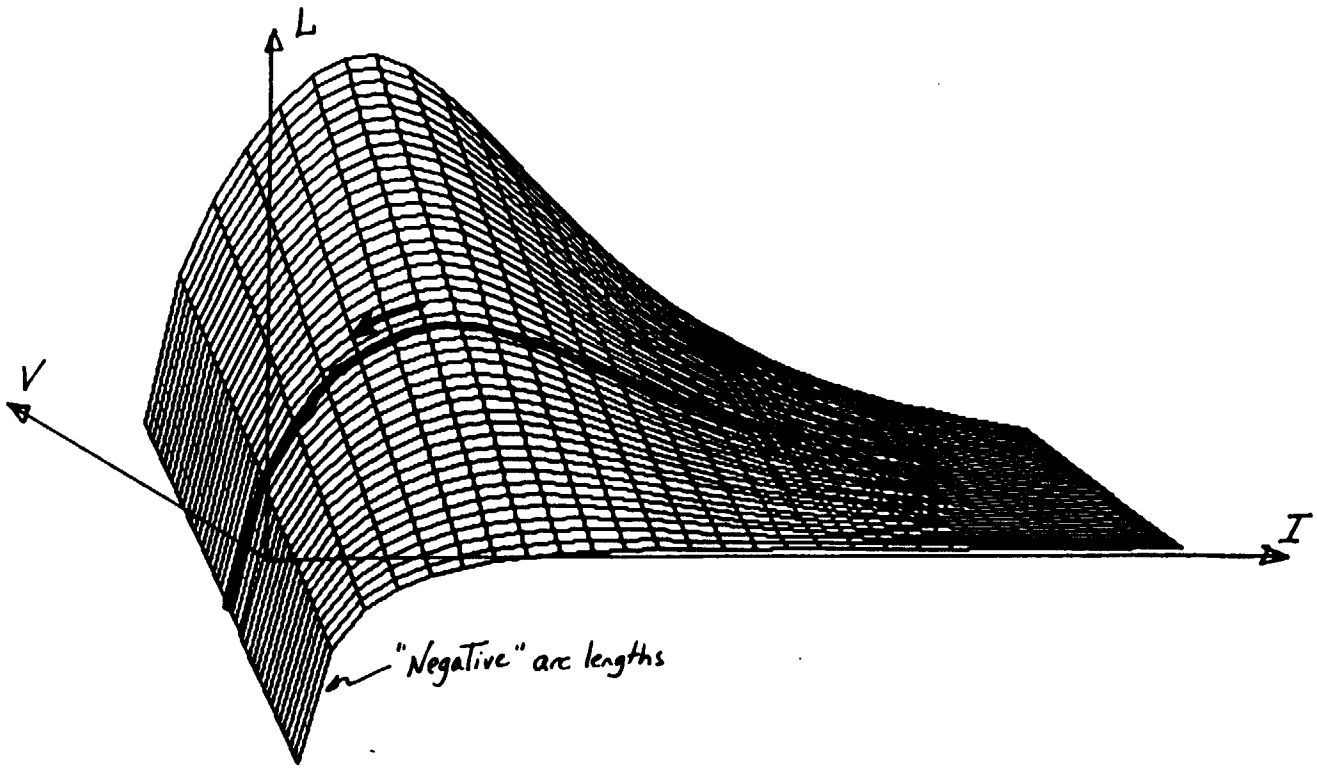


Figure 27. A three-dimensional view of arc length as a function of arc voltage and current. A fixed-voltage curve is highlighted.

tion results in the least tendency for crater cracking, microfissuring, etc. In general, the power density of the arc on the pool determines the instantaneous heat input to the metal, and thus it can be used to control the cooling rate of the pool. The cooling rate, in turn, is an instrumental factor in controlling the tendency to weld cracking.

Approach (1) above is the one which is inherently implemented in conventional AVC systems. If the goal is to keep the voltage constant while the current is ramped down from a nominal welding current the arc length typically increases in the beginning and then the electrode is driven back to the workpiece. Due to the physics of the arc this variation of the arc length is necessary to keep the arc voltage constant. As the current approaches zero, the arc length approaches zero as well and consequently the electrode tip runs into the solidifying weld pool. In this case the net power input to the pool (i.e., voltage times current, ignoring arc efficiency variations) is well defined. The power density, however, is related to the varying arc length and the dispersion of the arc column on the workpiece. Thus, the conventional AVC, while relatively simple, provides poor means for controlling the cooling rate of the solidifying weld pool. Furthermore, as described above and shown by the preceding Figures, it has an inherent tendency to drive the electrode into the solidifying weld pool.

If approach (2) is selected and the arc length is to be maintained at a fixed level regardless of current changes, the arc voltage has to vary correspondingly to keep the arc length fixed. In this case the electrode tip stays at a fixed, well-defined, height above the pool surface and thus the danger of it touching the pool is minimized. On the other hand, the total arc power (voltage times current) is not constant, because the arc voltage varies as the current varies.

A third approach, which is the combination of the two previously mentioned, is entirely possible with the adaptive AVC. Both arc voltage and length can be controlled to some degree for a given current. In this case, although the voltage determines the arc length, and vice versa, there are infinitely many possibilities of voltage-length combinations which can be reached for any current value.

It should be kept in mind that the two main goals of the adaptive AVC are to precisely control the heat input to the weld, particularly during weld termination when cracks and other defects tend to form, and at the same time prevent the electrode from touching the weld pool or the workpiece. Tight control of heat input permits improved control of weld cooling rate, which in turn affects the grain growth and the tendency for crack and microfissure formation. Clearly the constant-voltage approach, discussed above, fails in that it drives the electrode tip into the weld pool during current rampdown. Disabling the AVC, as commonly practiced, prevents this catastrophe, but instead the control over arc heat input is lost. Only through use of the second approach (controlled arc length) or combination of the first and sec-

ond approaches (simultaneously controlled arc length and voltage), is it possible to control the heat input to the pool as the current is ramped down, until the arc is terminated. Preliminary tests, combined with common welding experience, indicate that moving the electrode towards the workpiece, as the current is ramped down, is likely to be a key to diminished cracking during current tailout. To support this. a number of welding researchers have pointed out that keeping the arc density constant over the shrinking arc area, as the current is decreased, is highly advantageous for minimizing cracking and microfissuring. Once the capabilities of the digital AVC are available, experimentations with various arc length-voltage trajectories can be carried out to determine the techniques which result in minimum defects.

5.2.3 The Theory Of Automatic Voltage Control

In this section, the basic characteristics of the traditional AVC system are summarized. The improvements necessary to implement true arc length control, and to make the AVC robust enough to be operational during current tailout, are discussed as well.

Typically the torch height is maintained by a DC servomotor where the angular position of its shaft, θ_m, is related to the input voltage, V_m, by the transfer function G_m:

$$G_m = \frac{\theta_m}{V_m} = \frac{K_m}{s\,(s + a_m)} \qquad (14)$$

where K_m and a_m are constants characterizing the physical properties of the servomotor and therefore not modifiable (Figure 28). A power amplifier with gain K_a is usually employed to drive the motor. Furthermore, velocity feedback is typically provided from a tachometer or by sampled back-emf motor voltage for further control, through a feedback gain K_t. Finally, a gear assembly converts the angular displacement, θ_m, into a linear torch height displacement, L, by a factor K_l. Thus, the forward servosystem transfer function (incorporating the velocity feedback) becomes:

$$G_s = \frac{L}{V_s} = \frac{K_a K_m K_l}{s\,(s + a_m + K_a K_m K_l K_t)} = \frac{K_s}{s\,(s + a_s)} \qquad (15)$$

Note that both the transient and steady-state characteristics of this system can be adjusted by K_a and K_t. Thus, unlike the previous transfer function, where K_m and a_m are fixed, this new transfer function allows tuning of both K_s and a_s.

In a typical AVC system the arc length (sometimes referred to as arc gap distance), L, is indirectly sensed through the arc voltage, V_{arc}. In the design of such a

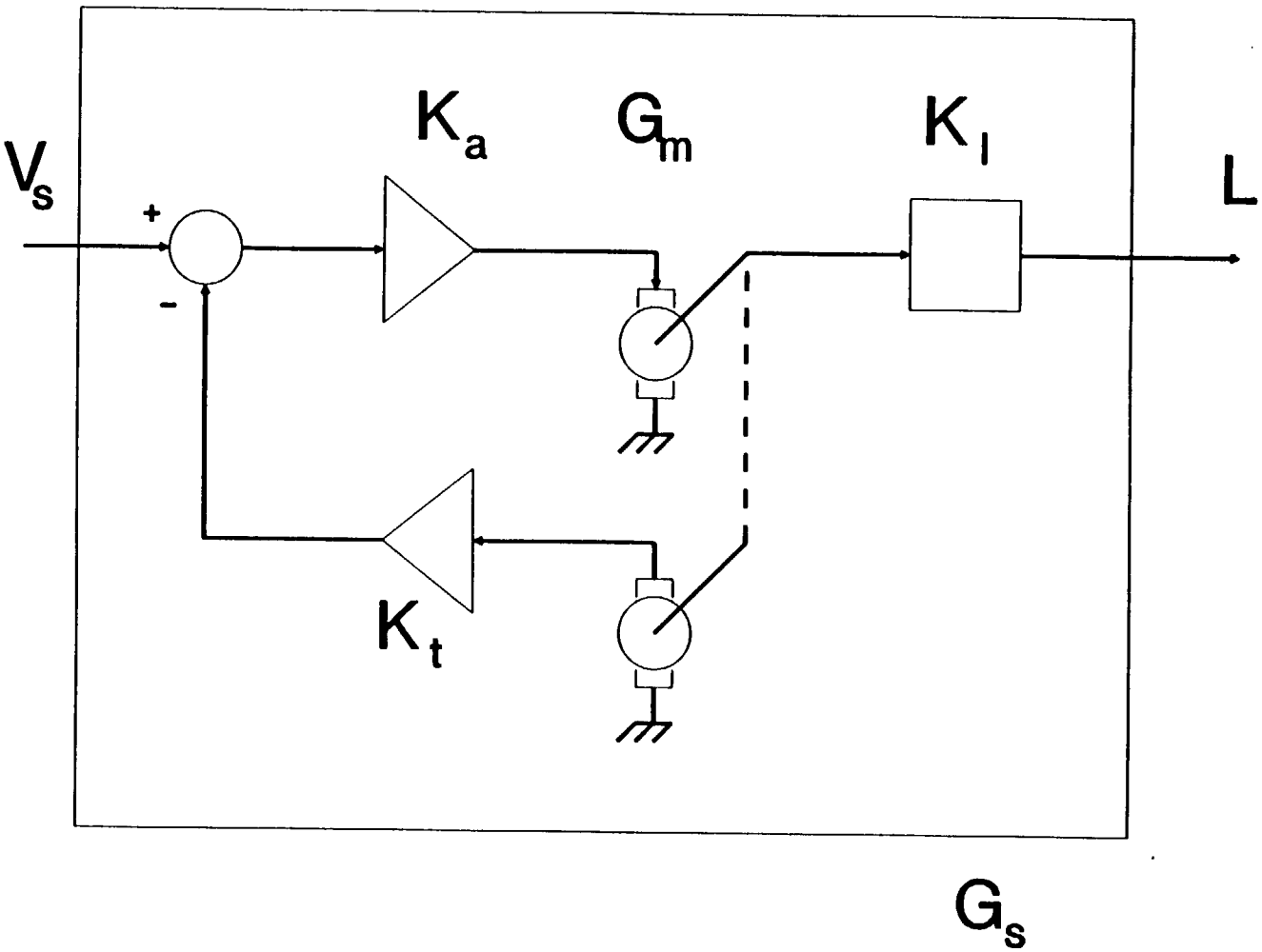


Figure 28. A servosystem controlling the arc length (torch height), L, through the input voltage, V_s. The total servosystem transfer function is G_s.

system it is usually assumed that the arc voltage varies linearly with arc length and therefore $K_v = dV_{arc}/dL$ is constant. It has been pointed out in the Phase I report of this research, however, that this is not the case. Specifically, it is important to note that (i) for any constant current level V_{arc}/L is highly nonlinear at short arc lengths, and (ii) $K_v = dV_{arc}/dL$ varies substantially with current. In the following discussion it will be shown that because of these arc characteristics traditional AVC systems, designed with the assumption of linear $V_{arc}(L)$, usually perform inadequately during current upsloping and downsloping periods. Refer to Figure 29 where the block K_v denotes the variable arc voltage-to-arc length sensitivity. The dynamic transfer function of the AVC is now given as:

$$G_{avc} = \frac{V_{arc}}{V_{ref}} = \frac{K_s K_v}{s^2 + s a_s + K_s K_v} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (16)$$

i.e., a second order system with a damping ratio ζ, and an undamped natural frequency ω_n. Most general texts on classical control theory, such as Ogata [38], discuss how the parameters ζ and ω_n determine system step response with respect to rise time, peak time, overshoot, etc. Critical damping is obtained when ζ is unity and any value of ζ less than 1 results in an underdamped system which overshoots in response to a step and oscillates until it settles to equilibrium. A value of ζ larger than 1 makes the system overdamped, which is usually undesirable because of the increased sluggishness. For an underdamped system the amount of overshoot as a fraction of the total input step size is found as

$$M_p = exp\,(-\pi\zeta/\sqrt{(1-\zeta^2)}) \quad (17)$$

The damping ratio is given as $\zeta = a_s/\omega_n = a_s/\sqrt{(K_s K_v)}$. If an AVC system is designed for critical damping at a given value of K_v, any increase in this parameter will reduce the damping ratio and give rise to an overshooting step response of the AVC. Furthermore, decaying oscillations of an approximate frequency of $\omega_n = \sqrt{(K_s K_v)}$ will occur as well. The percentage amount of step response overshoot as K_v is increased beyond the value resulting in critical damping (normalized at 1) is illustrated in Figure 30. The result of this is that increase in K_v beyond its optimal value, during welding, will make the AVC servosystem prone to overshooting and oscillations when noise perturbations, which in practice are unavoidable, occur in the system.

As the arc current approaches zero, the AVC will force the electrode tip closer to the workpiece surface. Theoretically, however, the AVC should be able to maintain a constant arc length when the current is constant, although the length may be very small for relatively low currents. In practice this is not the case, however, due to the variable arc sensitivity, K_v, which was discussed in the previous subsection. There it was shown that at low currents the AVC system becomes under-

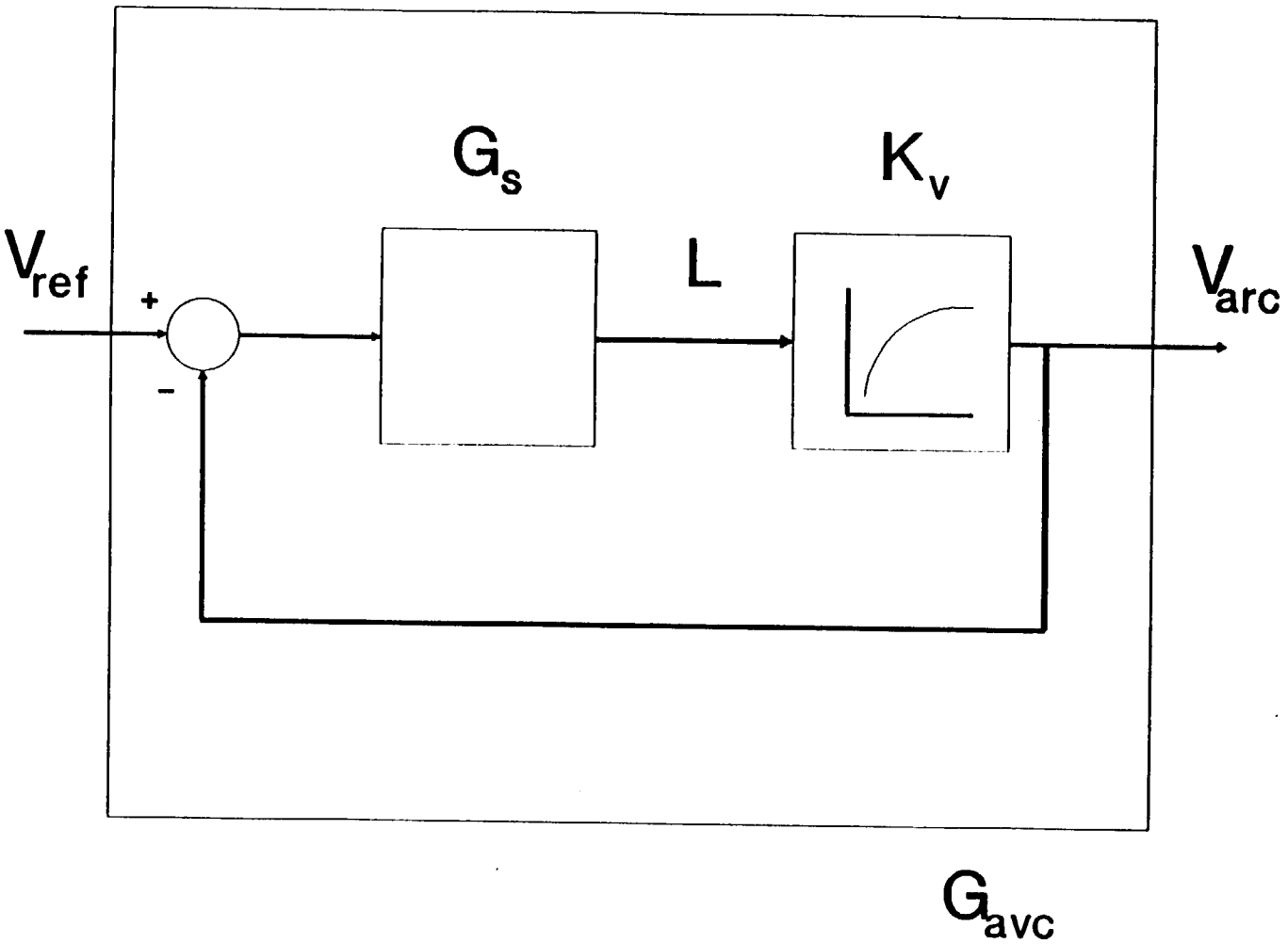


Figure 29. The AVC system uses the arc voltage, V_{arc}, as a feedback signal to be maintained at the reference voltage, V_{ref}. K_v indicates the nonlinear, current-dependent, V_{arc}-L relationship.

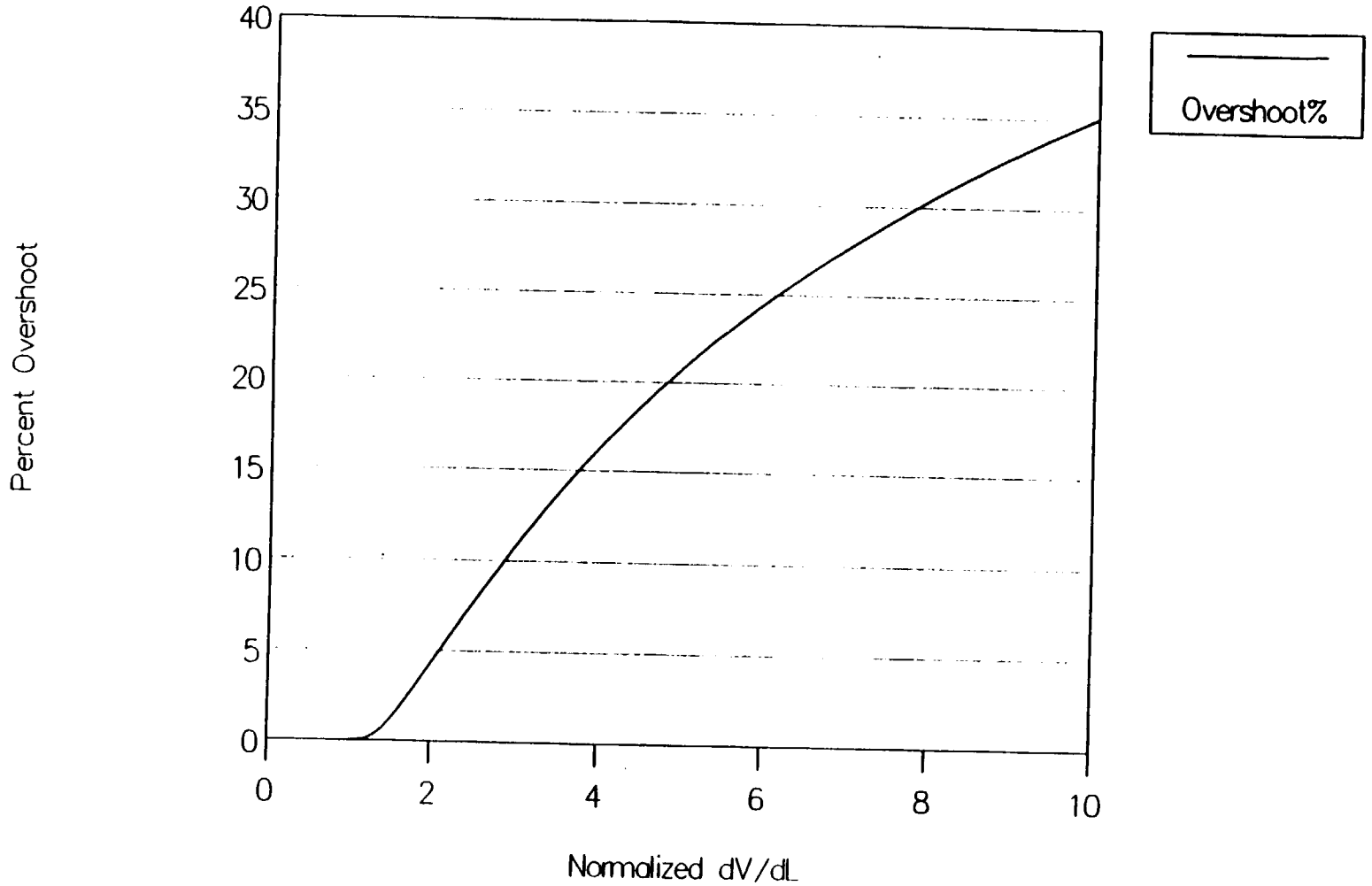


Figure 30. AVC step response overshoot as a function of K_v.

damped (because of increased K_v). Any perturbations in the arc (electrical noise, pool surface variations, etc.), which in practice are inevitable, cause the electrode to reposition itself to maintain constant voltage. Because of the underdamping of the system such corrective actions, however slight they may be required to be, will take place with substantial overshoot. For an electrode tip already very close to the weld pool surface this overshoot is frequently sufficient to make the electrode and the pool touch, which immediately extinguishes the arc. In other cases the electrode may be at a safe distance from the pool surface. With sufficient underdamping arc perturbations will cause the torch to oscillate at the system resonance frequency, ω_n, which is a common low-current AVC behavior.

Variations in the arc sensitivity, K_v, can also degrade the performance of the AVC in the opposite way, i.e., make it overdamped instead of underdamped. If the nominal current range for which the AVC was designed is not the one yielding the lowest K_v the system will become more sluggish as K_v decreases. This problem can be just as serious as the overshoots and oscillations of the underdamped system. The AVC must be able to follow all significant surface and pool variations during the weld pass, rendering too slow an AVC unacceptable.

Because the dynamics of the AVC system are a primary concern in its design, specific experiments were carried out to examine the validity of the common assumption that this factor is constant. An arc was maintained on a copper plate and the arc sensitivity d_{Varc}/d_{Larc} measured as a function of current and voltage for two shielding gases. The gases were argon and helium, flowing at 658 and 752cm^3/min (35 and 40 cfh) respectively. The included angle of the Tungsten electrode was 24 degrees and the arc length was 2.38mm (3/32"). The arc sensitivity as a function of current for arc voltages of 12, 13, and 14V is plotted in Figures 31 through 33 respectively. Argon was used as a shielding gas. Note that generally the arc sensitivity K_v ranges approximately from 0.45 to 1.4V/mm. For helium gas the arc sensitivity was measured in the same way for voltages of 26 and 28V and the results are shown in Figures 34 and 35, respectively. Figure 36 shows the results at both voltages together and here the arc sensitivity varies from about 1.2 to 6.0V/mm.

The above measurements reveal that the arc sensitivity K_v is considerably affected by current, voltage, and shield gas. Therefore the response of the arc voltage control system varies accordingly with the affecting variables. A simulation of a representative AVC system was carried out to study the significance of the variable arc sensitivity. The results are illustrated in Figures 37 and 38. Figure 37 shows how the arc voltage responses to step changes in the reference voltage where argon is used for shield gas. The system constants were designed for optimum response at 100A. At 300A current, however, the arc sensitivity K_v has varied enough to invalidate the optimum response design and as a result the arc voltage (and correspondingly the arc length or TWD) exhibits an overshoot of about 20% before it settles to

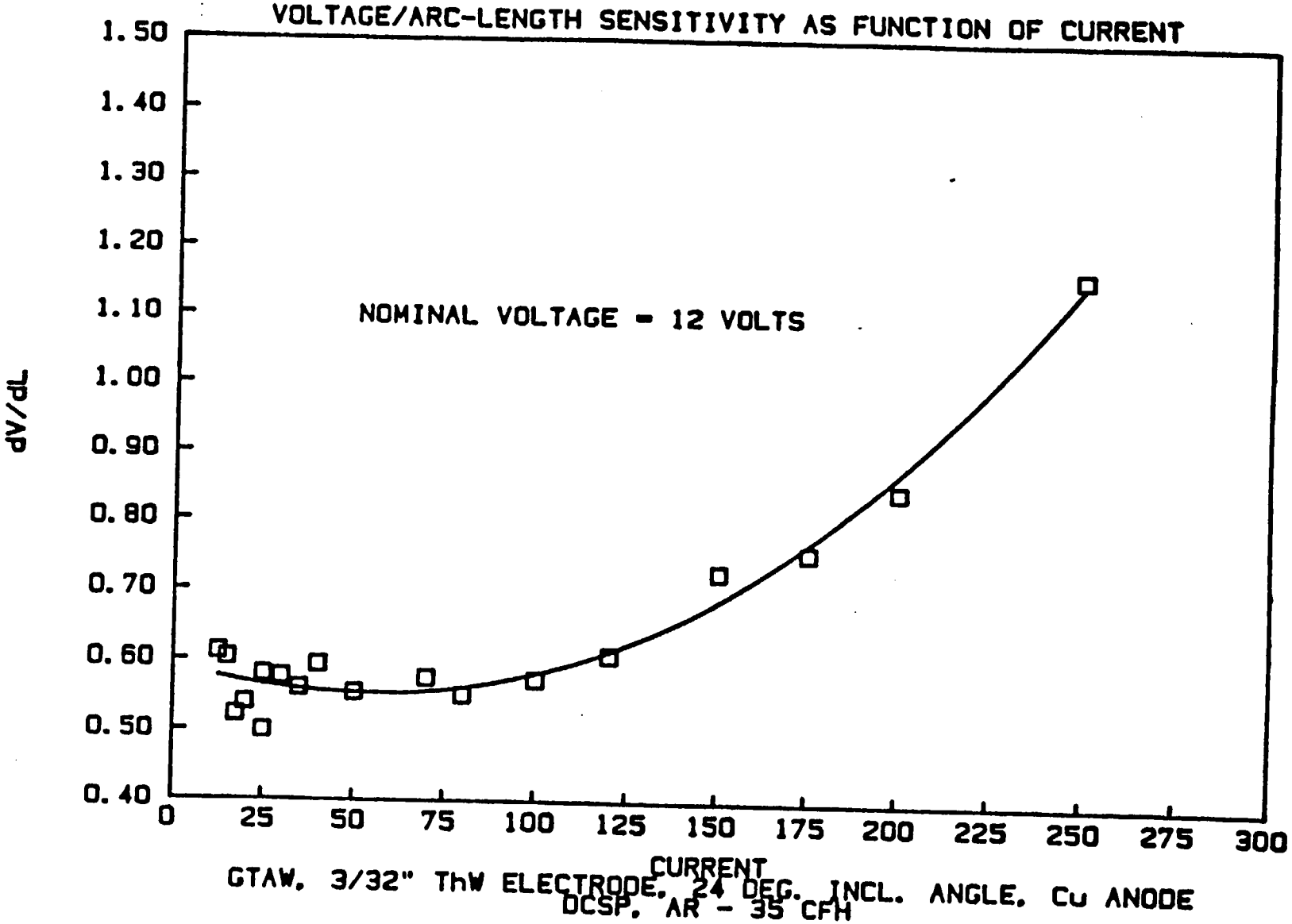


Figure 31. Arc voltage-to-length sensitivity as a function of current. Arc voltage is 12.0 V.

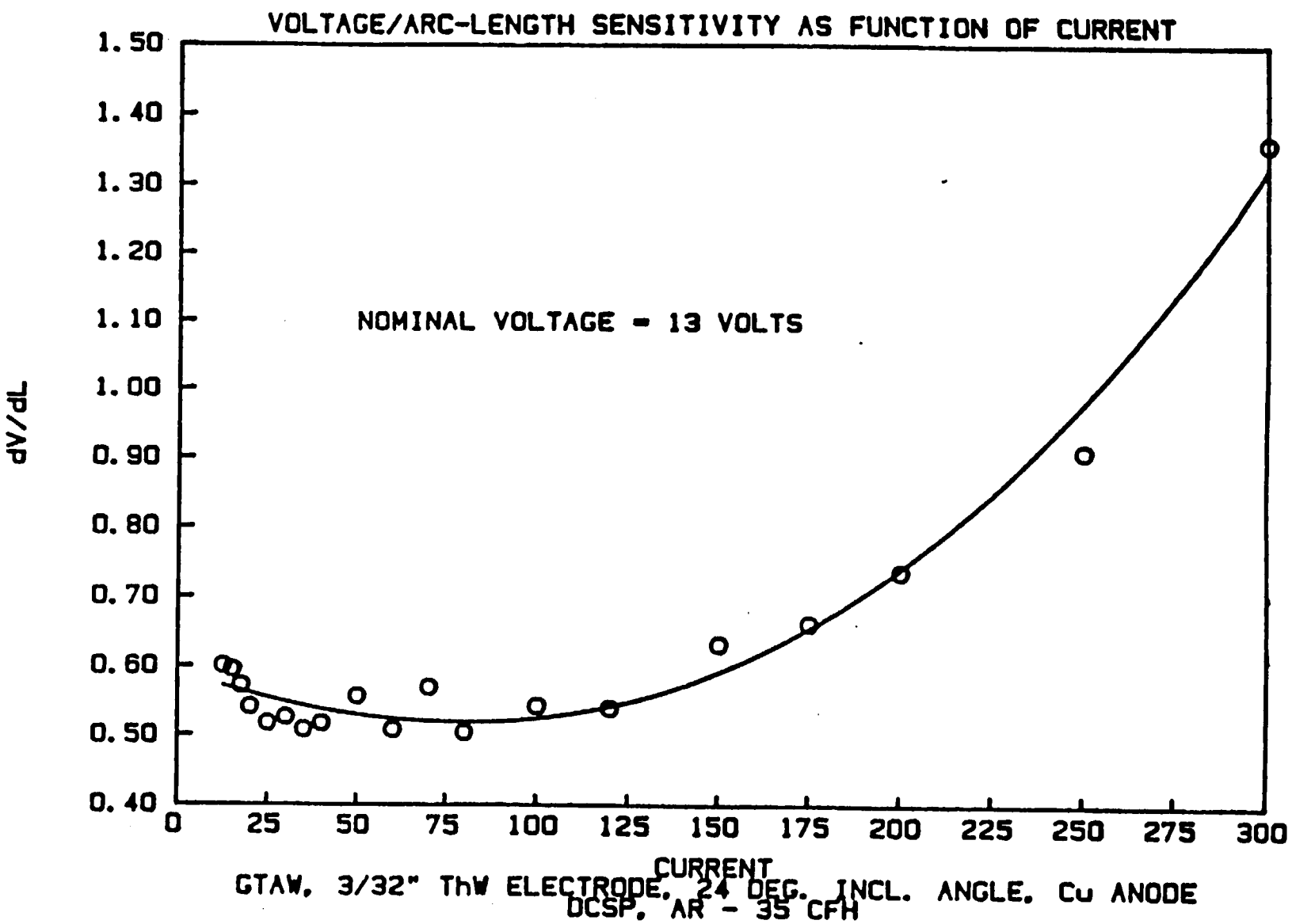


Figure 32. Arc voltage-to-length sensitivity as a function of current. Arc voltage is 13.0 V.

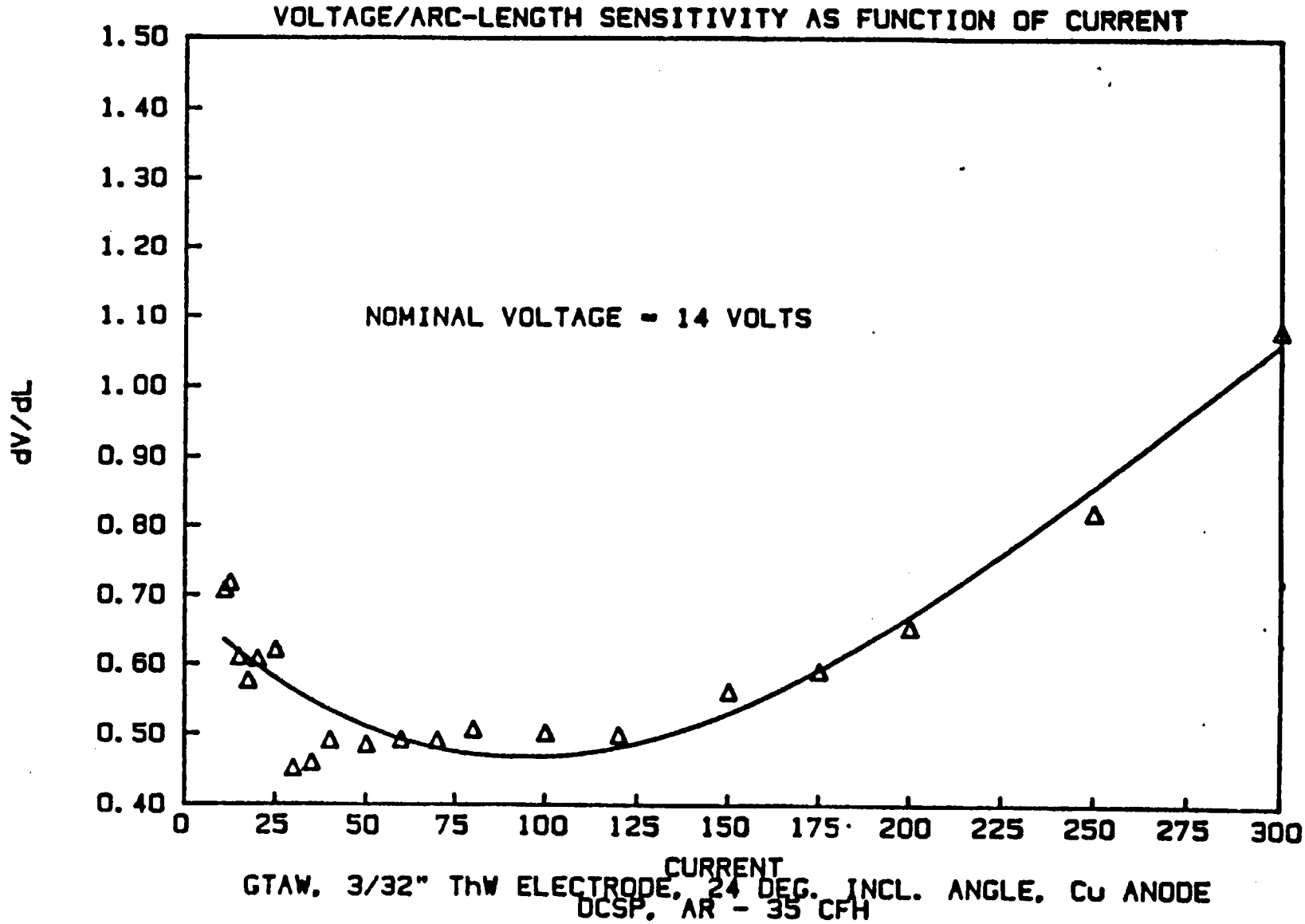


Figure 33. Arc voltage-to-length sensitivity as a function of current. Arc voltage is 14.0 V.

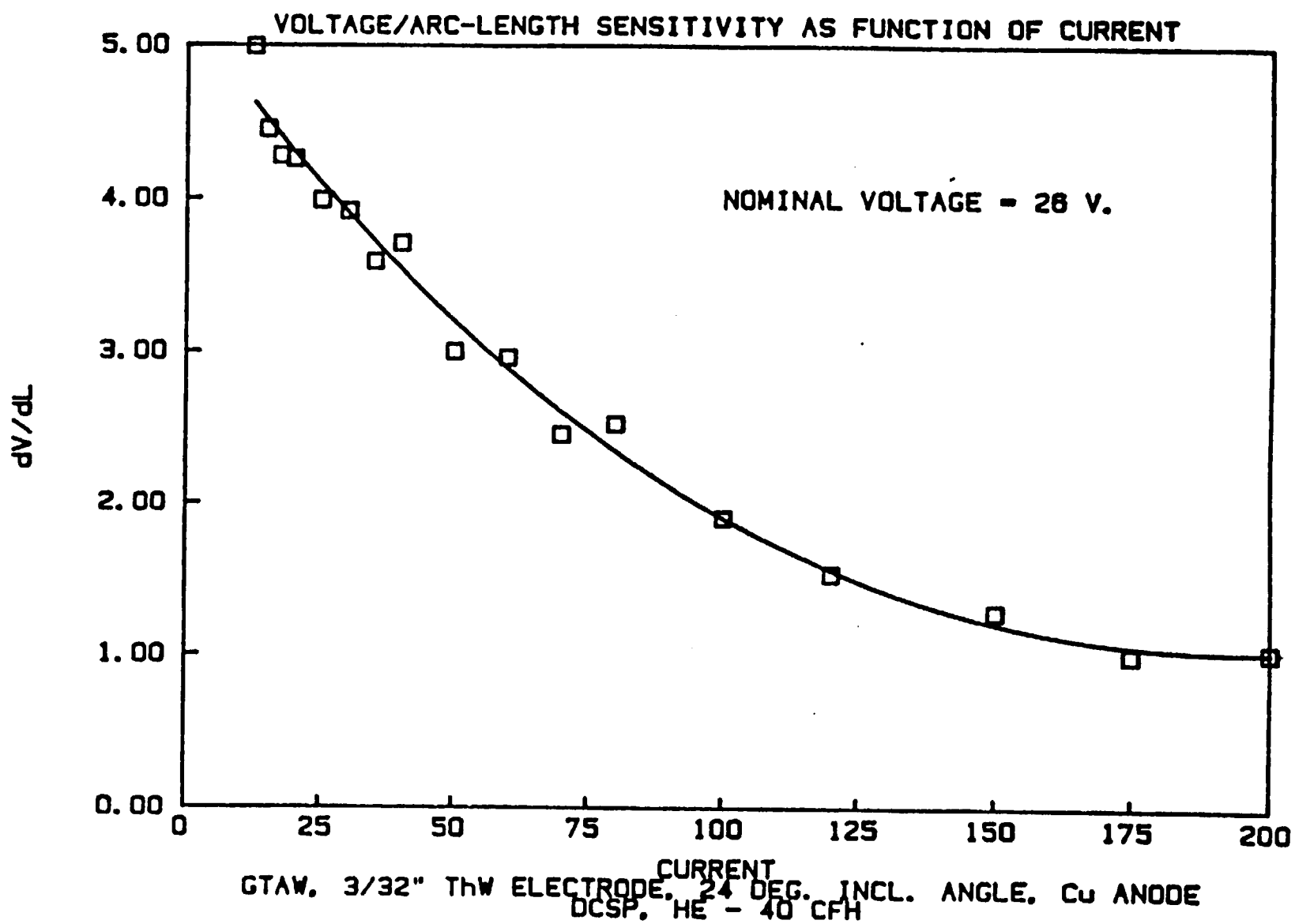


Figure 34. Arc voltage-to-length sensitivity as a function of current. Arc voltage is 26.0 V.

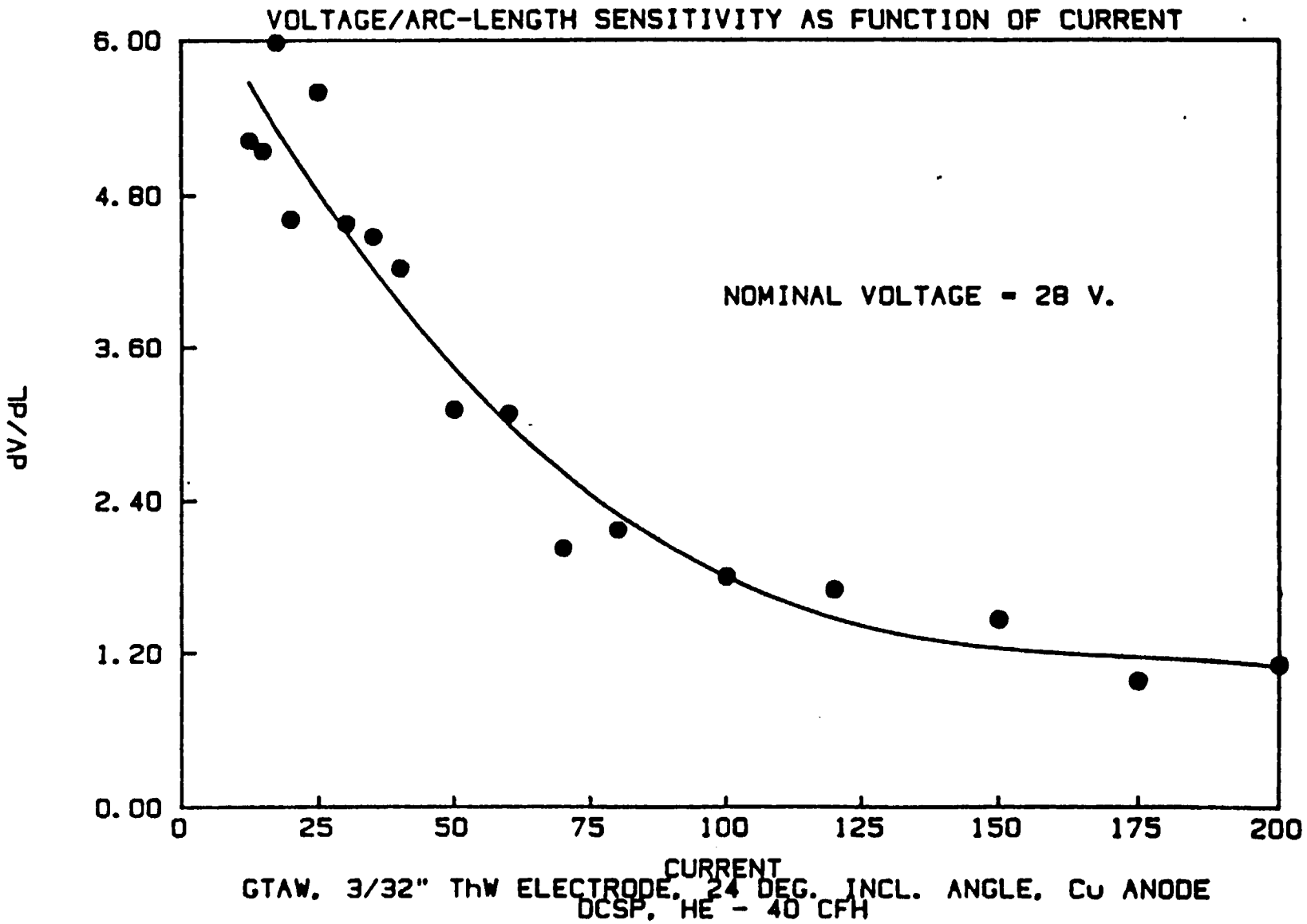


Figure 35. Arc voltage-to-length sensitivity as a function of current. Arc voltage is 28.0 V

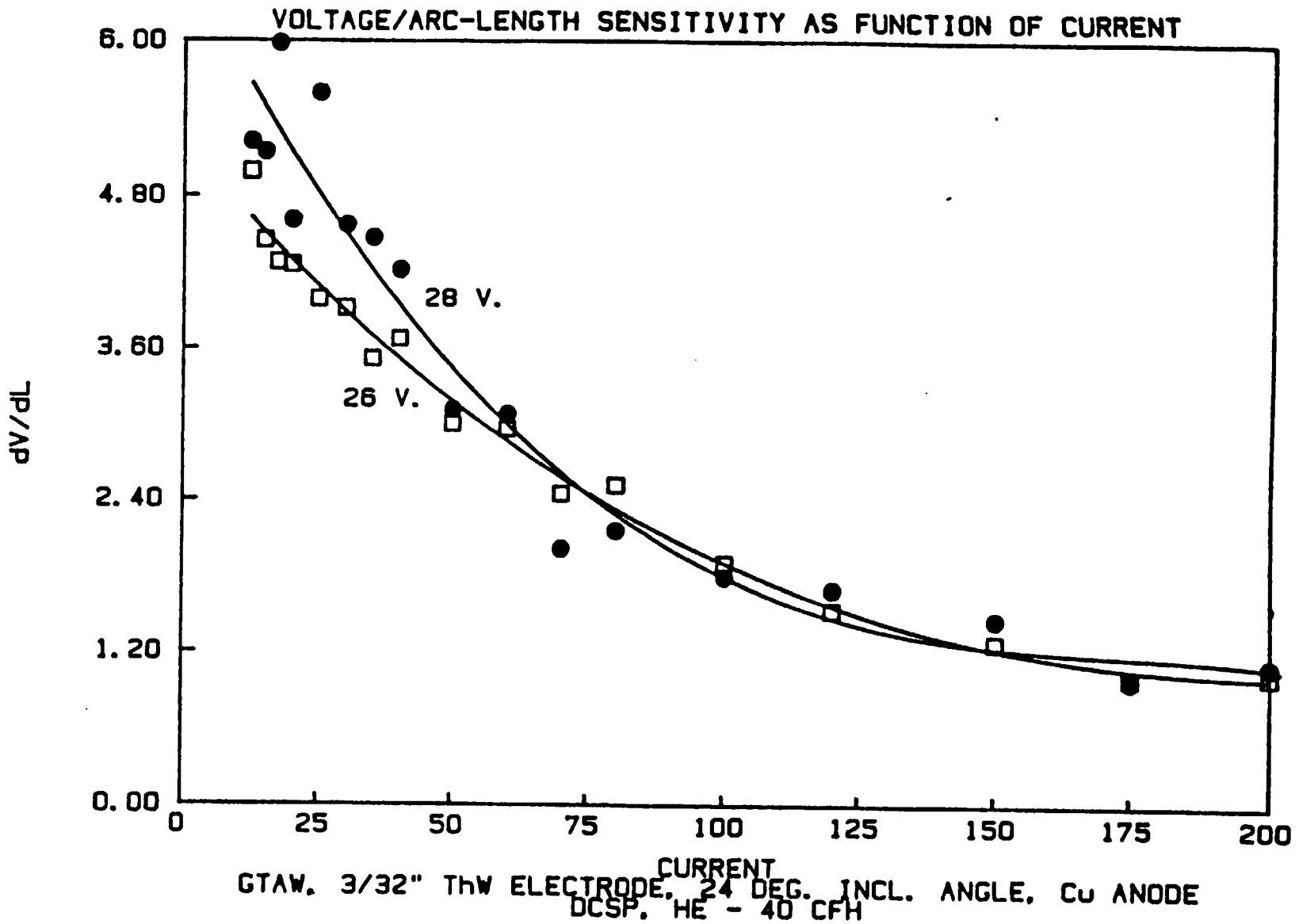
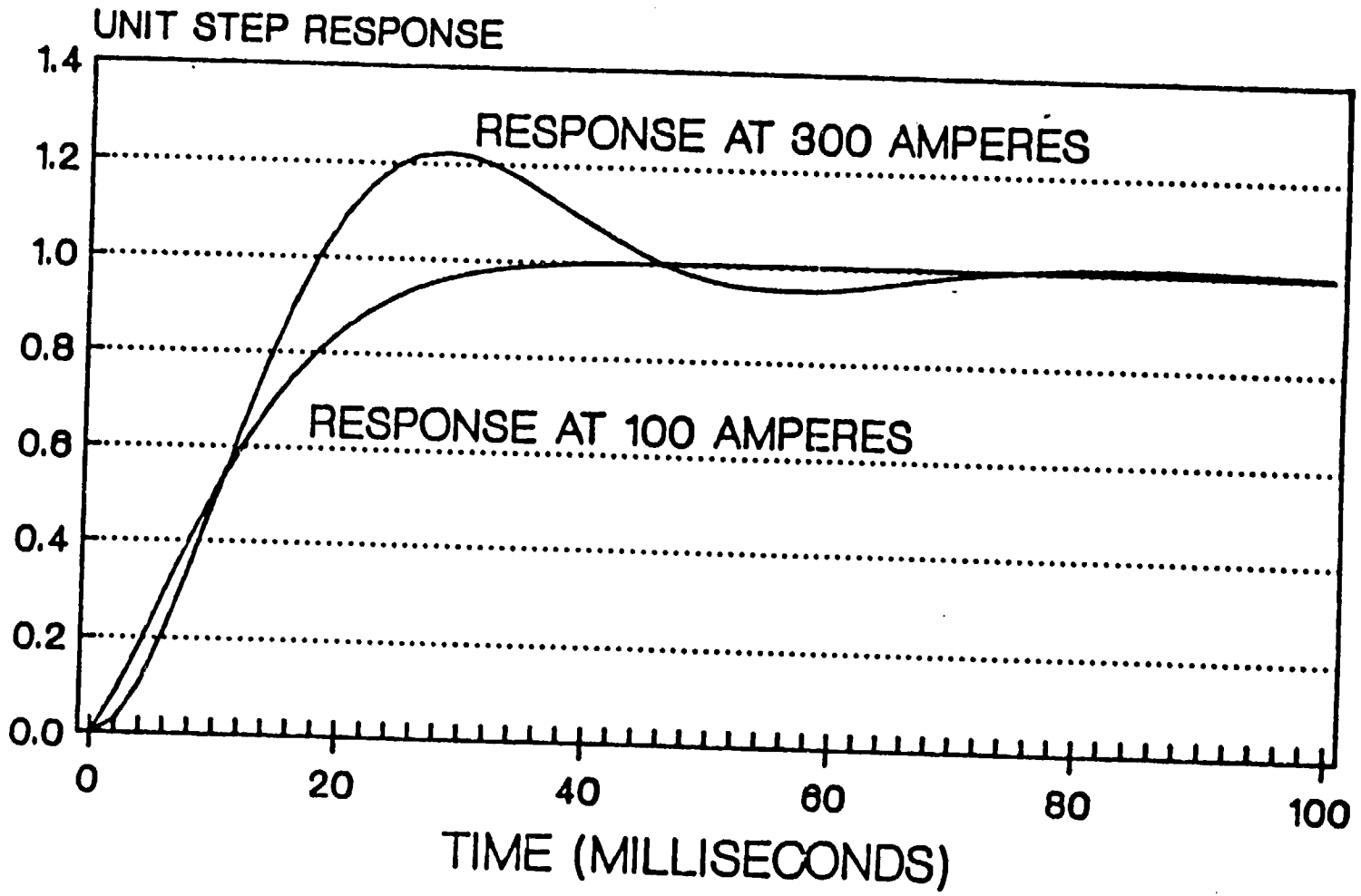


Figure 36. Arc voltage-to-length sensitivity as a function of current for helium. Arc voltage is 26 and 28 Volts.

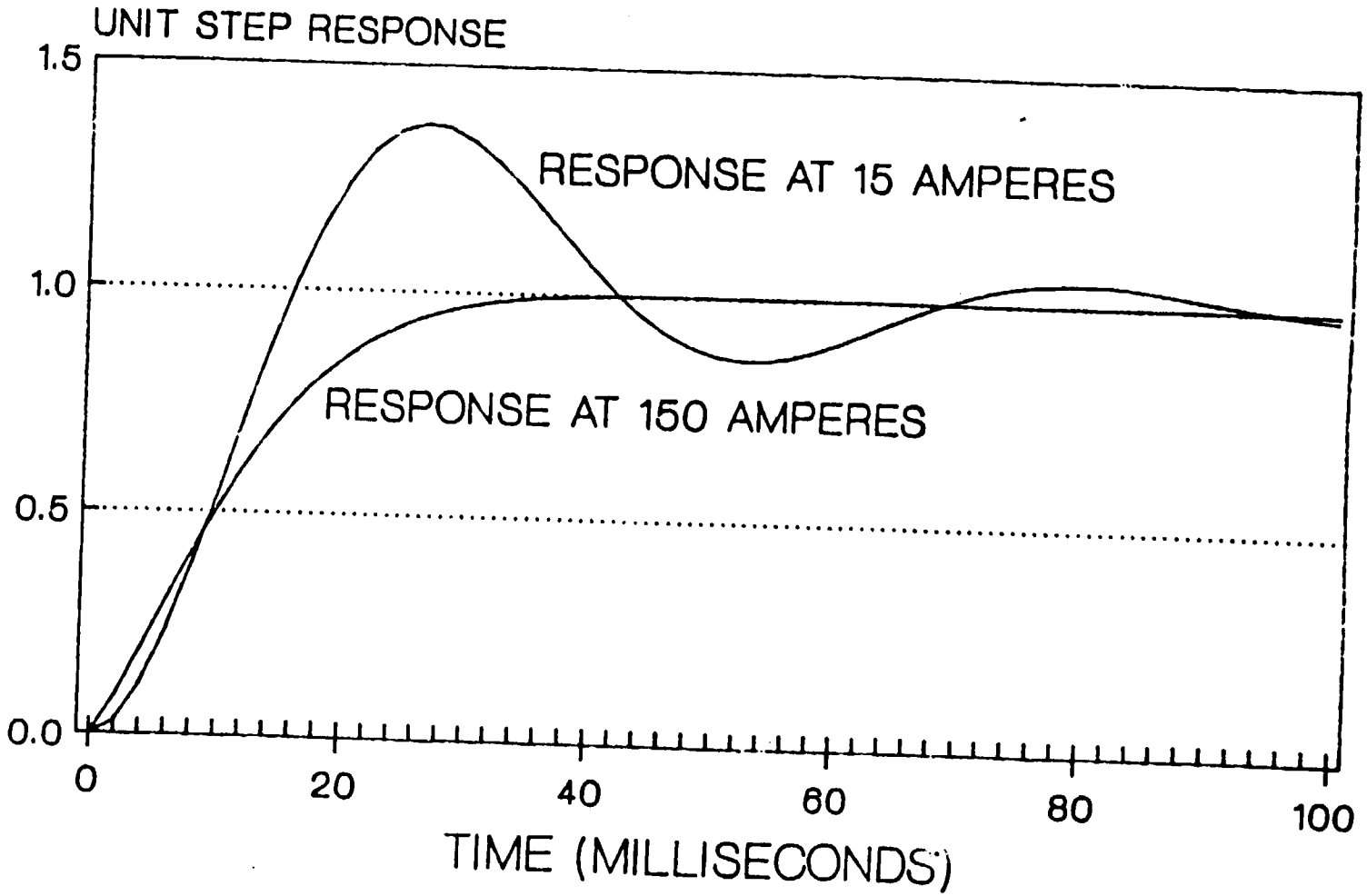
AVC RESPONSE



AR - 35 CFH, NOMINAL VOLTAGE = 13 V.

Figure 37. Arc voltage response to current steps at 100 and 300 Amperes. Shielding gas is Argon. This system is not optimally responsive at 100 Amperes.

AVC RESPONSE



HE - 40 CFH, NOMINAL VOLTAGE = 26 V.

Figure 38. Arc voltage response to current steps of 100 Amperes. Shielding gas is Helium. Note that the system is not optimally responsive at 150 Amperes.

its final value. A similar test was carried out for the helium case (Figure 38), where the system was optimized for 150A and then the arc voltage response was examined at a current of 15A. In this event the overshoot was about 40%. Overshoots of this magnitude and larger occur in actual welding situations when the current is ramped up or down through low current levels (e.g. during startup or termination). To prevent driving the electrode into the workpiece (during upslope and downslope) the AVC is frequently turned off temporarily, as classical voltage controllers can not maintain stable control under those conditions. Both simulation examples above demonstrate the need for compensation of varying arc sensitivity if optimal arc voltage and length response is to be maintained for all welding conditions.

The capability of voltage or arc length control during tailout is important for continuous control of power input during current downsloping, which in turn affects the cooling rate of the weld and its susceptibility to cracking. The usual approach of turning the AVC off during downsloping may result in an initial decrease in voltage before the voltage starts increasing to maintain the arc across the fixed gap distance. The net result of power input is shown in Figure 39. Here, a welding current of 125 A is assumed, and a gap length of 0.030". As the welding current is linearly sloped down from 125 A down to 5 A the arc voltage changes from about 9 V down to just over 6 V, and then it rises up to about 14 V. This is visualized by tracking the voltage curve from right to left (the current is decreasing). Multiplying the varying voltage by the decreasing current yields the total power delivered to the torch, dropping from just over 1 KW down to less than 100 W. Although the arc current decreases linearly the power decreases in a nonlinear fashion as shown. The time history of power input affects the weld cooling rate. Through the proposed AVC modifications the weld power can be controlled at all times, thus allowing the welder to affect the cooling process at the initiation and termination of the weld pass.

5.2.4 Implementation Of The Digital AVC

Three versions of the improved AVC were implemented for this research. The first one was constructed using conventional analog circuit design, with the exception of a variable-gain amplifier, which could be controlled from an external computer. The purpose of the variable-gain amplifier was to enable the computer to adjust the system gain so as to keep the damping ratio optimal at all current levels. A block diagram of this modified AVC system is illustrated in Figure 40. Insertions have been placed into the conventional analog control loop in two places. In the feedforward path a multiplying DAC (digital-to-analog converter), such as the AD7528, is added for adaptive control of the feedforward gain. Nominally, the gain of this inserted block is close to 1, leaving the AVC in its original configuration. Its gain is be varied, however, as the arc sensitivity, K_v, varies due to the welding current. The desired gain of the DAC is found from the arc sensitivity data. In essence

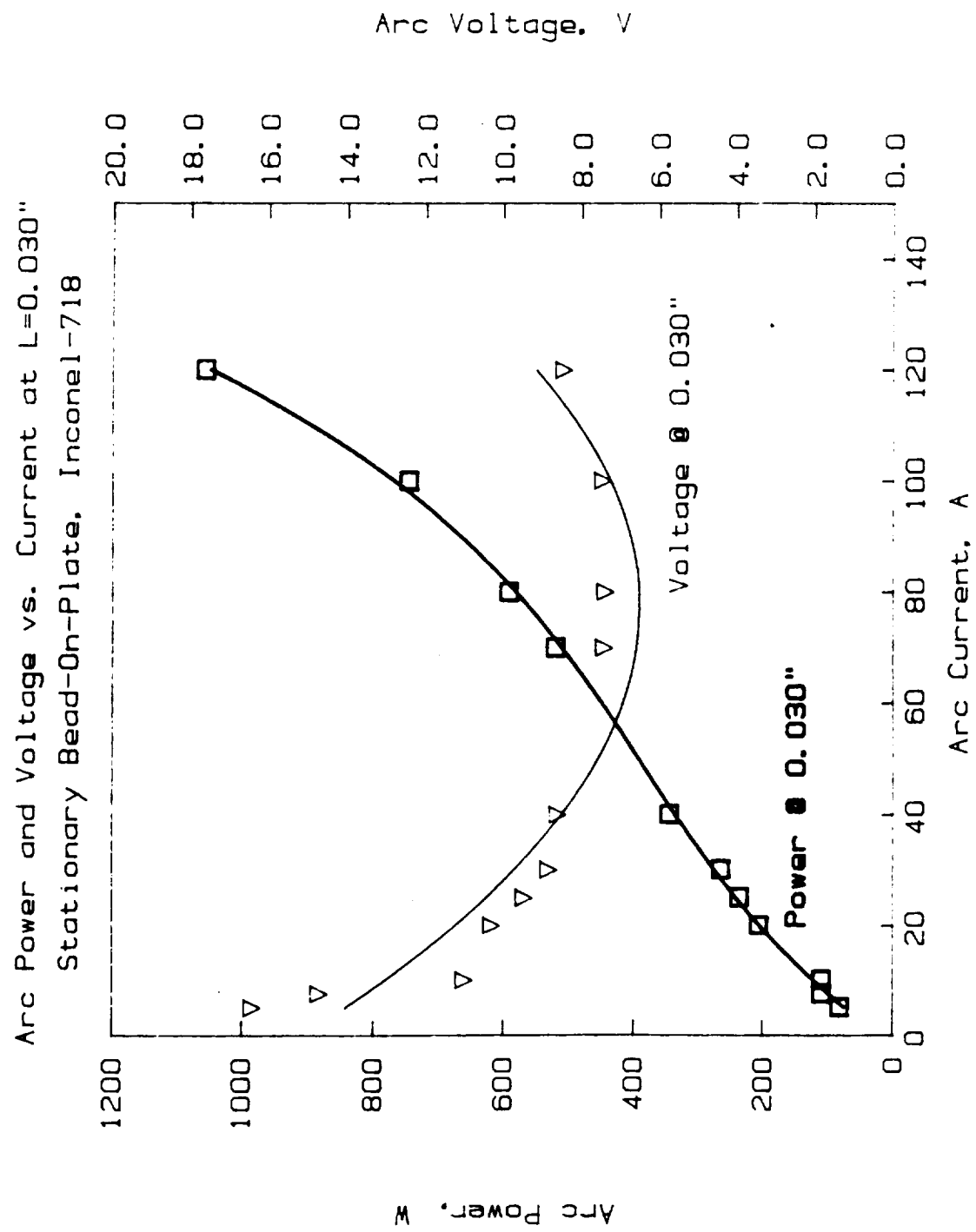


Figure 39. Arc power and voltage displayed as a function of current. Power varies nonlinearly as current is ramped down.

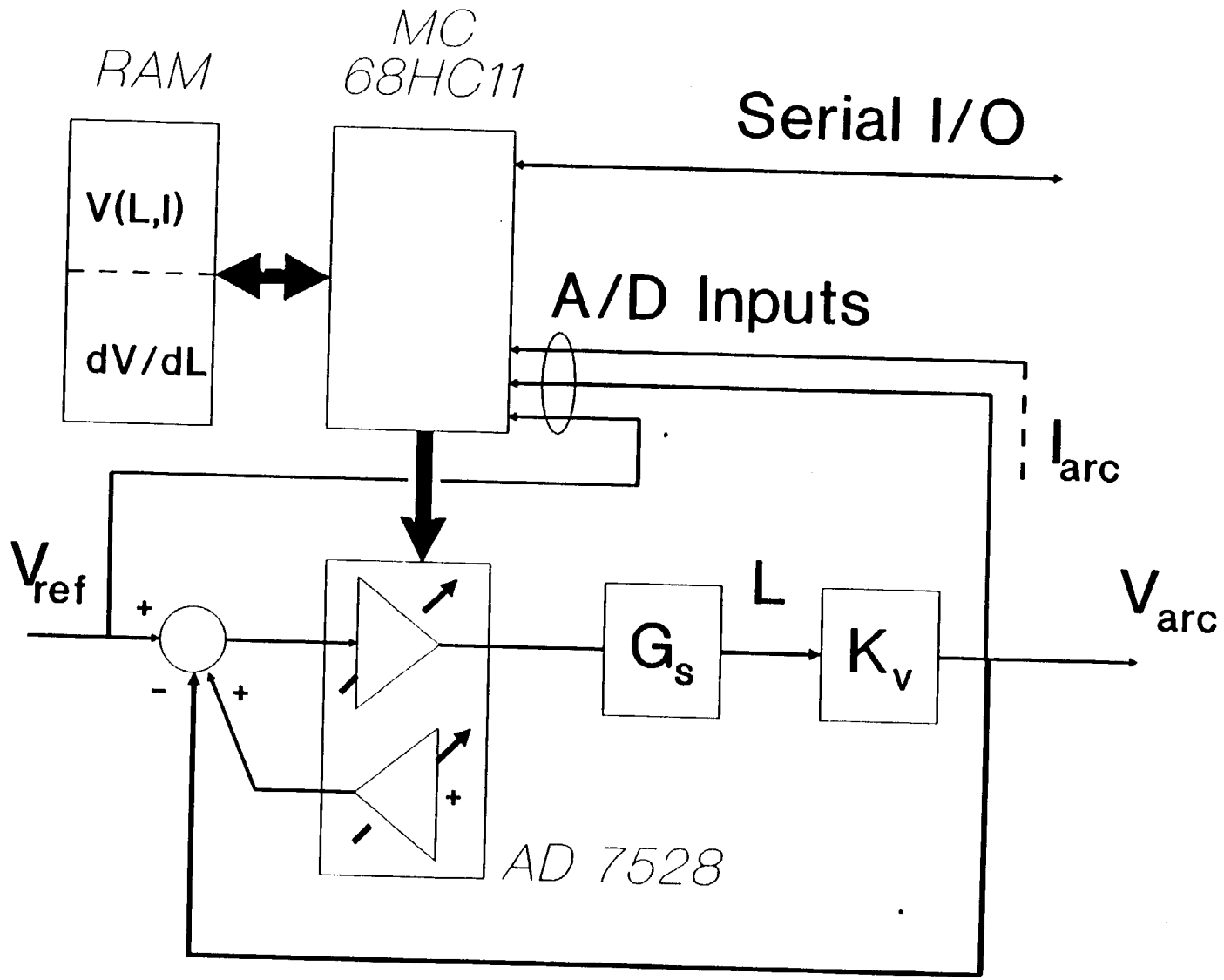


Figure 40. The modified AVC. An adaptive gain amplifier is increased into the forward path. The feedback voltage is modified at the summing point.

the variations of K_v are countered by the variable DAC gain, leaving the net feedforward gain of the loop constant at its optimum value. This keeps the AVC damping characteristic at optimum at all current levels and prevents excessive sluggishness on one hand and underdamped oscillations on the other. To enable arc gap limitation or arc gap control an additional summing, implemented at the existing summing junction of the present AVC, was necessary. The output from the second half of the dual DAC is added to the feedback signal, resulting in a "modified" feedback signal. During normal welding the DAC output is zero, leaving the feedback voltage unaltered. Arc gap limiting is achieved by injecting the difference between the required and the actual arc voltage ($V_{ref} - V_{arc}$) from the DAC into the summing junction. The result is that the net feedback voltage will always equal the reference voltage, and therefore no error signal will be generated to move the AVC servomechanism. Arc gap control can be achieved by the same circuitry, but now the arc voltage required for the desired gap length is continuously calculated and the DAC output is selected so as to maintain this gap. Specifically, the voltage from the DAC, V_{DAC}, in this case is $(V_{ref} - V_{gap})$ where V_{gap} is the voltage required for the desired gap distance, as found from the RAM table.

This circuit was built and tested to a limited extent. The arc voltage, current, and length were measured by a data acquisition system and corresponding tables or maps were constructed from these measurements. These maps were for (i) voltage vs. current and arc length, and (ii) voltage sensitivity to arc length (i.e., dV/dL) vs. arc current and arc voltage. The initial maps were constructed by sweeping the arc length and arc current over typical welding ranges. Details of the actual control block prototype are sketched in Figure 41. The reference voltage (which may be a DC setpoint voltage or a pulsed reference signal) is subtracted from the arc voltage (during system calibration the arc length, obtained from an LVDT is used for this purpose, however). Furthermore, the contribution from the servosystem tachometer is added to this summing junction. The resulting signal is amplified and entered to the MC1495 circuit, which is an analog multiplier. Essentially, the difference between output pins 2 and 14 is proportional to the multiple of the voltages on input pins 9 and 4. Pin 4 is therefore used as a gain control input in this circuit, and its voltage is either determined by an external source (a computer) or internally (with jumper JP1 is connected). The purpose of operational amplifier U8 is to convert the differential signal from the analog multiplier to a single ended signal. The resulting signal is amplified and routed to a power amplifier which drives the torch servomotor.

The damping ratio of the system could be varied through external commands, which controlled the gain of the MC1495. During testing, however, it was determined that a control loop implemented in software on the VME-based computer, which would be delivered to NASA, would be adequately fast and yet simpler in im-

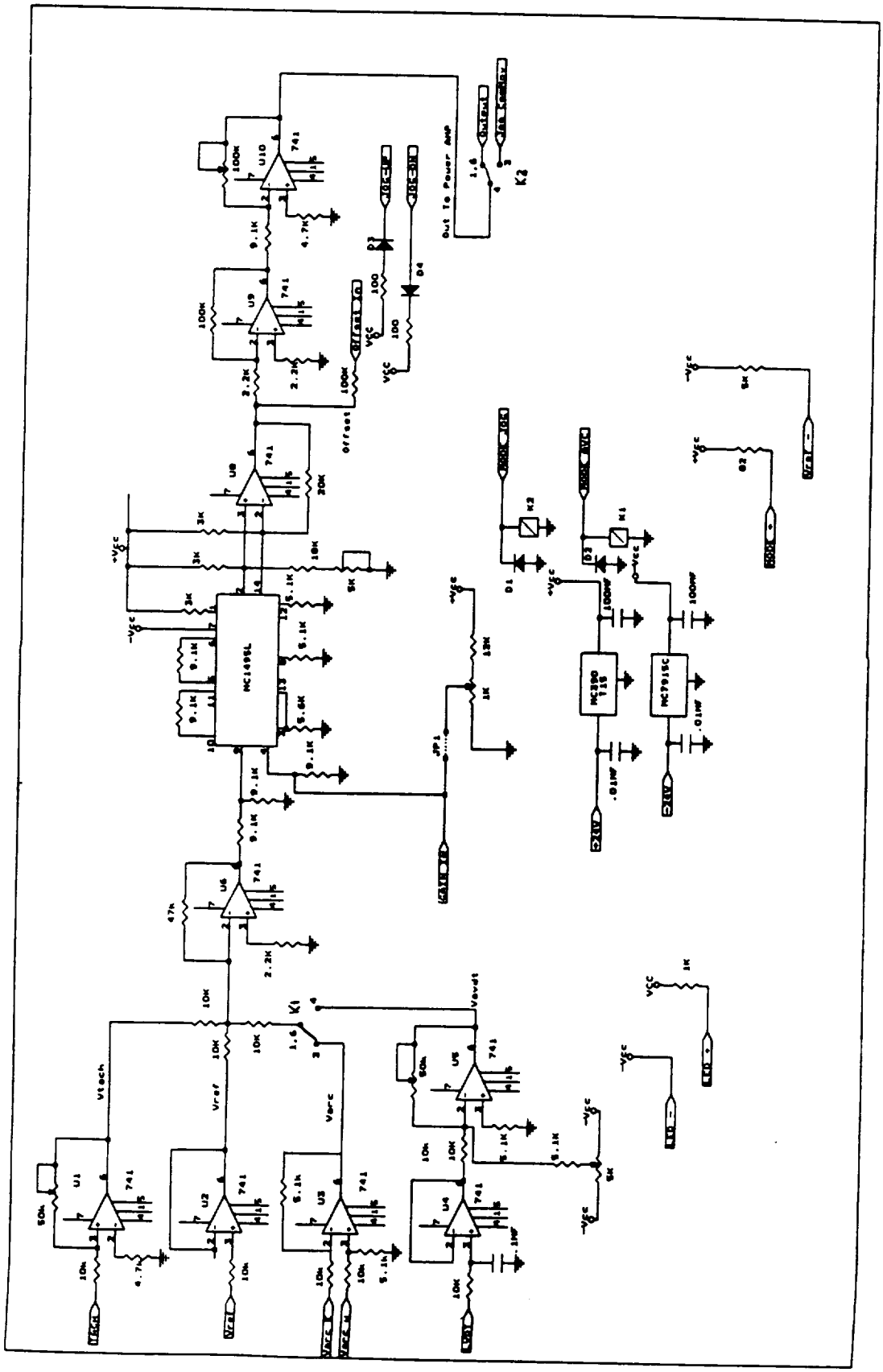


Figure 41. A schematic illustration of the adaptive AVC.

plementation. Thus, further refinements of this circuitry were abandoned, but the experience gathered through its use was carried on to the next AVC generation.

A preliminary software-implemented AVC was constructed on an IBM-AT compatible computer. A software loop was implemented, which repeatedly sampled the arc voltage through an ADC (analog-digital converter), subtracted this voltage from the desired arc voltage, multiplied the difference by a gain constant, and sent the resulting value out to a DAC (digital-analog converter) port. The resulting analog voltage activated the motor to move the torch electrode either up or down until the arc voltage equaled the desired arc voltage. The AVC was demonstrated to the NASA personnel who visited Mid-South Engineering during the research period. Three control options were demonstrated using the digital AVC. First, the AVC was operated as a traditional AVC system and a constant voltage was maintained throughout the current ramp-down interval. The arc current was established at 160 Amperes and the arc voltage was set at 10.8 Volts. Once the welding procedure had reached steady-state, the current was ramped down to zero in about 13 seconds (this relatively long ramp-down period was selected to facilitate better observations of events as the current was reduced). Figure 42 shows the arc current, voltage, and relative arc length during the downsloping period. As the current ramps down, the arc voltage is essentially constant, as expected with the traditional AVC intact. Any variations in the arc voltage at this point are due to the dead band width programmed in the system. To maintain the constant arc voltage the AVC must first increase the arc gap and then rapidly decrease it. As the current approaches zero the arc gap diminishes rapidly and finally the electrode tip runs into the workpiece, as illustrated in the Figure. This is what is commonly encountered with traditional AVC systems, and therefore they are usually disabled during current ramp-down periods. It should be noted that the real-time gain scheduling scheme of the AVC, which is designed to optimize the dynamics of the system, does not prevent the electrode from running into the workpiece. Maintaining the loop response optimized through gain scheduling only ensures that the system is neither overdamped (sluggish) nor underdamped (oscillatory or jittery) at any time.

Figure 43 shows the digital AVC response, again controlled through a current downsloping period, but this time the arc length is maintained at a fixed level and therefore the arc voltage is allowed to vary to accommodate the fixed arc length. It is emphasized that this is not the equivalent to simply turning the AVC off, as frequently practiced with traditional AVC systems during downsloping intervals. Turning the AVC off would not maintain a constant arc length if the workpiece surface was uneven or sloping with respect to the torch trajectory. This could be clearly demonstrated on a circumferential pipe weld, where the weld is terminated as the arc returns back to the beginning of the weld bead. Furthermore, it should be noted that no external transducer, such as a Linear Variable Differential Transformer (LVDT), is used to measure the torch height. In fact, such a scheme

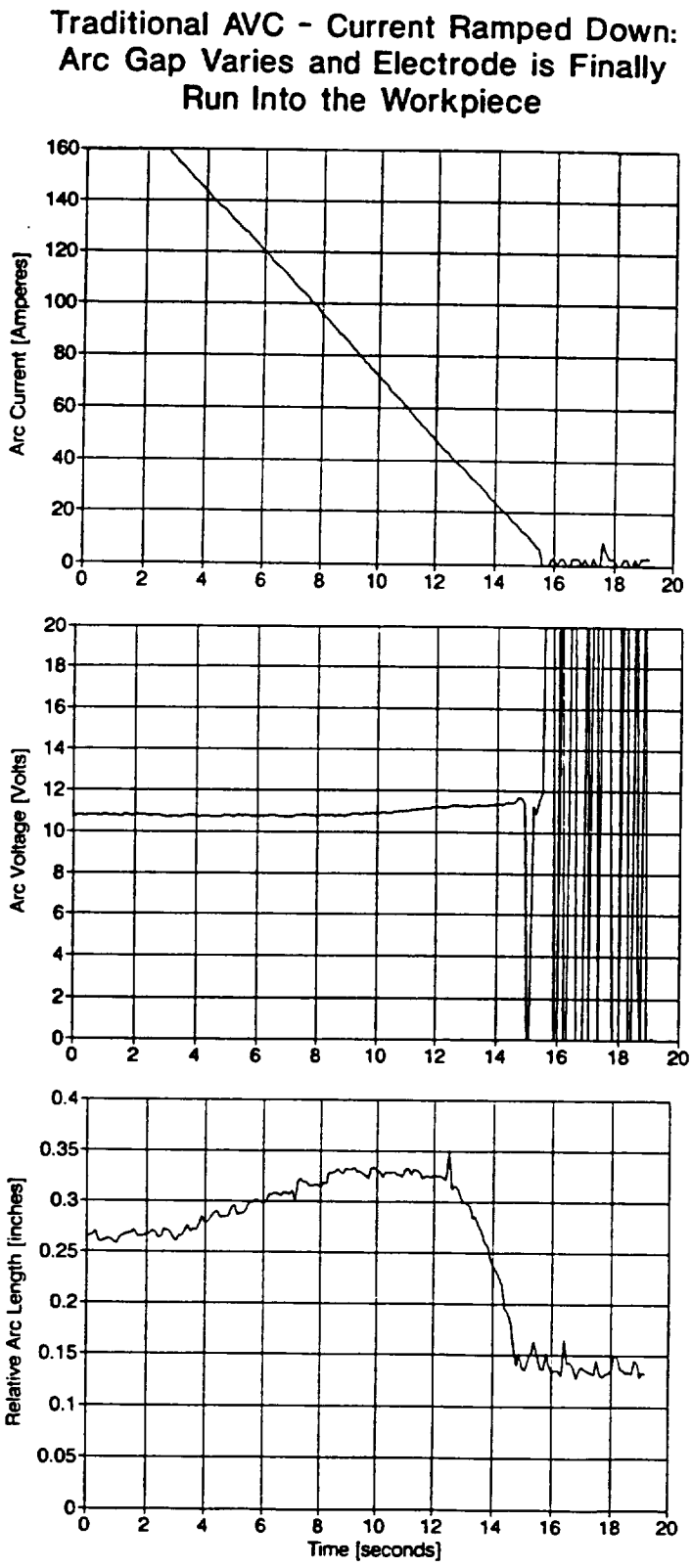


Figure 42. Traditional AVC - Current is ramped down: Arc gap varies and the electrode is run into the workpiece.

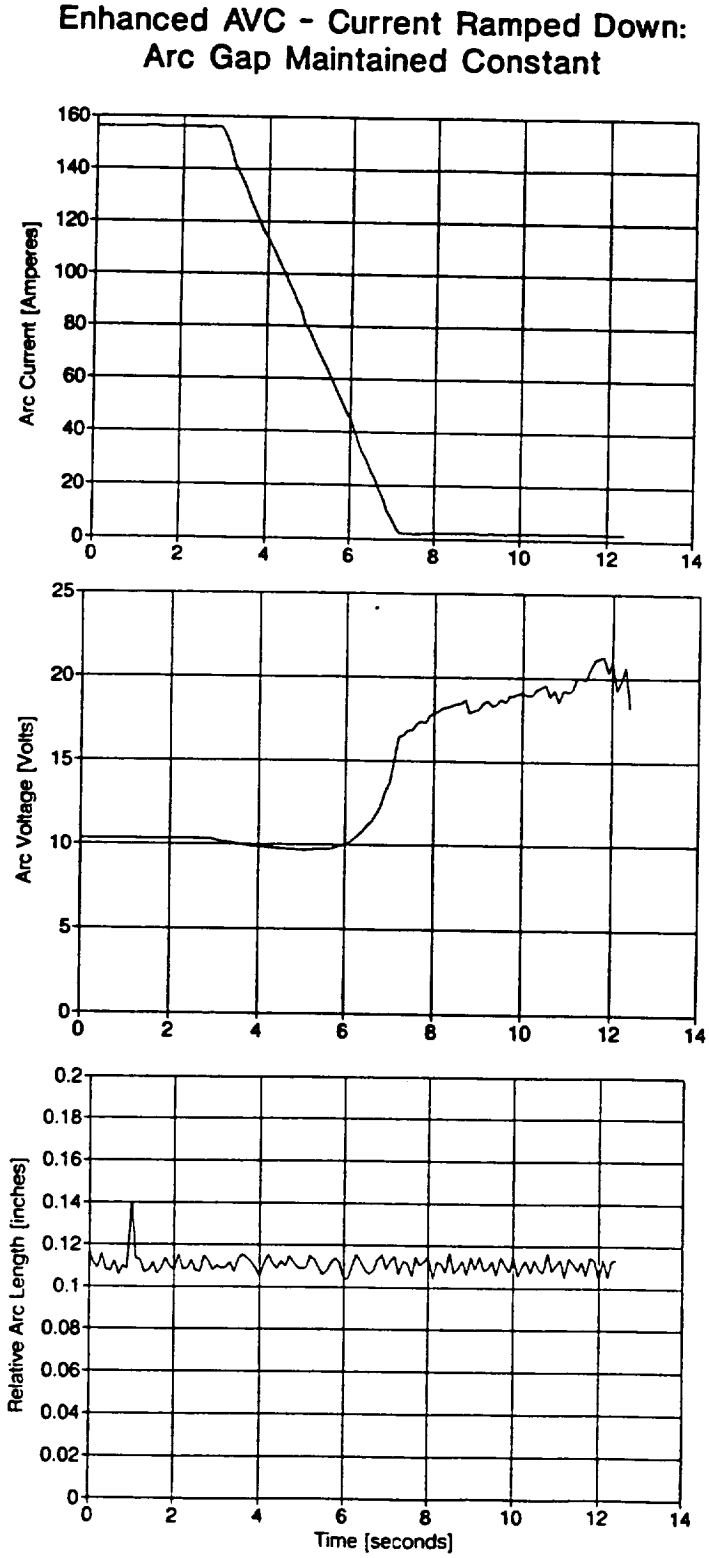


Figure 43. Enhanced AVC - Current is ramped down while the arc gap is maintained constant.

would not measure the distance between the electrode tip and the pool surface. Instead, the arc length is calculated here from the arc voltage-current-length maps shown earlier in this report. The Figure illustrates that the arc voltage must vary if constant arc length is to be maintained during current downsloping.

Finally, Figure 44 shows how the digital AVC can control the arc gap beyond simply maintaining a constant arc length during the downsloping period. In this case the arc length is varied in a controlled manner and reduced as the arc current is reduced to zero. As before, this control action would not be possible using an external displacement transducer mounted on the torch mechanism. Such a transducer would only enable control of the height of the torch with respect to the robot arm or fixture which holds the torch, but not the distance between the torch tip and the molten weld pool beneath it. The arc length control, illustrated in the Figure, is achieved through calculations based on the instantaneous arc voltage and current, and the controller adjusts the actual arc length until it equals the desired arc length value. The sampling rate of the digital AVC for these demonstrations was about 50 Hz, which is far more than adequate in view of the response time of the mechanical AVC servosystem.

The final AVC system, implemented on the VME-based computer, is also implemented in software and functions in a similar way as the IBM-AT version. The voltage sampling/update rate is currently set at 100Hz, but the maximum rate is about 500Hz. This maximum rate is limited by the complexity of computations that must be carried out during each sampling period. Implementing the AVC in software, rather than hardware, gives the designer greatly added flexibility in designing the control algorithm. The implementation of arc length control, in addition to voltage control, has already been discussed. Real-time adjustment of the control loop gain has been outlined as well. Design of the controller has facilitated experimentation with various configurations of PID control by simple software commands, and thus optimization of the AVC response is greatly simplified.

The Windows-based interface of the VME computer allows the user to select either voltage-control or length-control mode for the arc. In either case the user specifies the desired values for the selected variable, and the computer calculates the other variable, based on the mathematical relationships between arc voltage, length, and current. The arc length can be programmed to ramp up to a fixed level, and then decrease again as the current is tailed out. During the welding sequence, the arc voltage required for the programmed arc length and current is calculated and displayed on the screen. In addition to calculating the voltage or the arc length, the system varies the gain of the control loop so as to prevent instabilities during tailout.

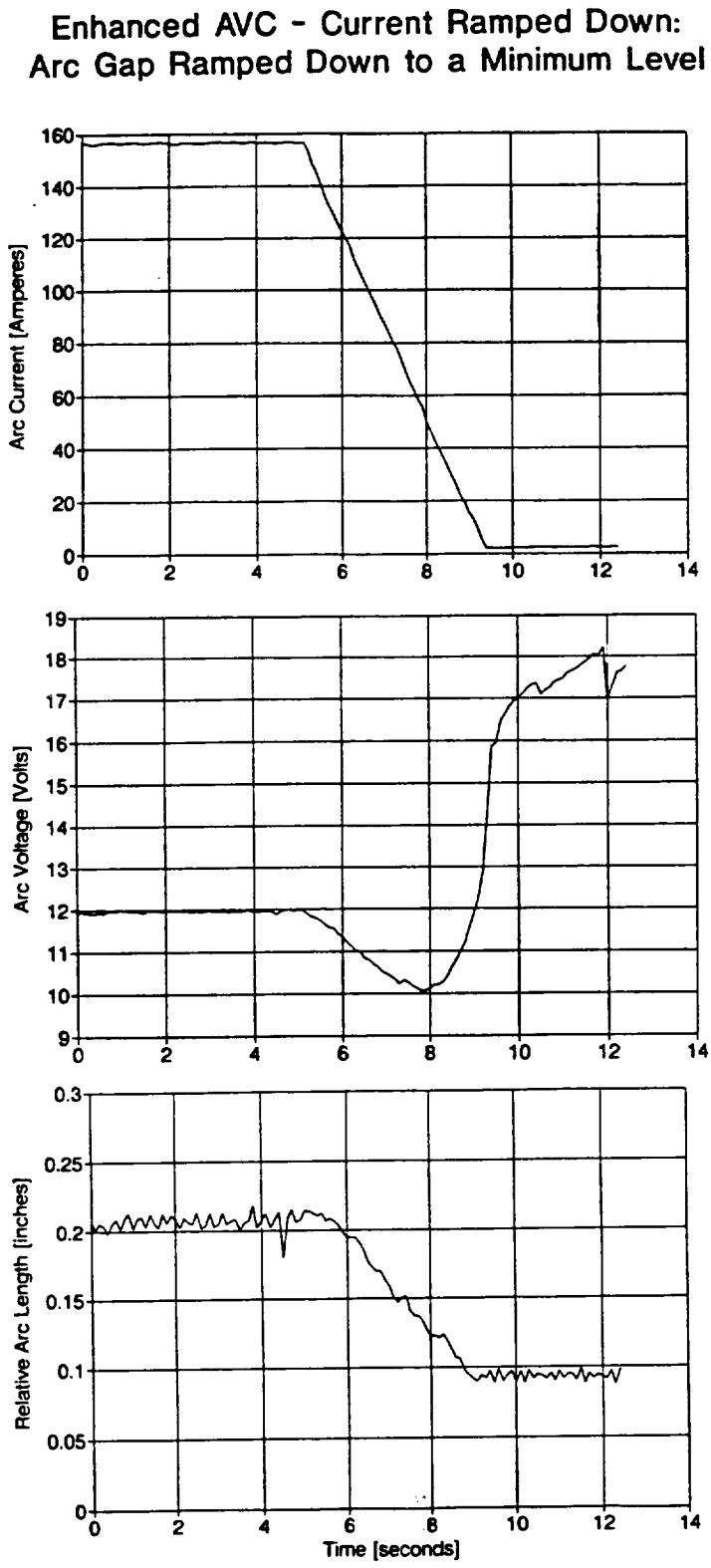


Figure 44. Enhanced AVC - Current is ramped down; Arc gap is ramped down as well.

5.3 Weld Parameter Control

The system designed during this research offers the user unique capabilities for determining the equipment variables necessary to achieve desired welding results. The primary components are the artificial neural network which aids the user in selecting the equipment parameters for the welding system, and the graphical weld parameter editor, which permits the user to define any arbitrary parameter variations throughout the weld graphically. These two constituents of the system controller are discussed in the following subsections.

5.3.1 Background

Complete control of all relevant indirect weld parameters (IWPs), or equipment parameters, of the process is imperative for high quality welding. For the GTAW process the primary parameters are usually considered as arc current, voltage, travel speed, and wire feed rate. The arc voltage is indirectly used to set the arc length, and the relationship between these two variables was discussed in the section on Automatic Voltage Control (5.2).

The artificial neural network methodology has been discussed in a previous section of this report. There a relatively small, 2-by-18 node network, was capable of predicting DWPs from given IWPs to an accuracy of on the order of 10%. The question of how reliable the neural network predictions are, has been brought up on numerous occasions. At this time no quantitative estimation methods exist that can answer this or similar questions. However, despite the rather exotic name, neural networks are well defined mathematical structures. In that sense a mapping neural network can be viewed as a nonlinear regression system, where each output variable is calculated as a nonlinear function of the input variables. Once the network has been trained, the nonlinear function yielding each output variable is determined and fixed. Each network function can be visualized as a continuous surface, interpolated among the data points in the training data. As a generalization, neural networks, like most other regression schemes, yield more accurate mapping as the data points are more "densely populated", and as the data is "better behaved" or less noisy.

For example, the density of data points, used to train a neural network for IWP-to-DWP mapping, can be increased by increasing the number of data points without expanding the ranges of welding current, voltage, travel speed, and wire feed rate. As the data point density increases, the overall accuracy of the network mapping will increase in the general region of training data values. Nothing can be asserted, however, about the neural network mapping accuracy, as predictions of weld parameters further from the general training set are attempted. On some occasions the predictions may be adequate, while on others they may fail. To avoid this, it is imperative that the training data set cover the regions of welding parame-

ters which are anticipated or most commonly used for welding. By using training data that yield generally acceptable and good welds, the neural network is implicitly biased to suggest welding parameters that fall into the range of acceptable values as well.

Continuing the analogy of neural networks to other regression schemes, it is important the training data is "well behaved", and contains as little noise or irregularities as possible. The GTAW process is by nature generally well behaved in the sense that, e.g., the relationship between any IWP and the DWPs is continuous and relatively smooth. For example, although the bead width dependence on arc current, voltage, travel speed, and wire feed rate is nonlinear, it is smooth and well behaved. To enable the neural network to model this relationship, it is important that the training data is similarly well behaved as well.

The equipment parameters predicted by the neural network from the desired weld geometry may be given back to either Nunes' or Tsai's weld models, which in turn yield their own estimates of weld geometry. Thus, a certain degree of verification of the combined performance of the neural network output and the weld models is possible.

The equipment parameters determined by the neural network, or otherwise determined by the user, are programmed into the system in a graphical form. This permits the user to define arbitrary values and variations of each of the primary equipment parameters: current, arc length, voltage, travel speed, and wire feed rate. The graphical editor used for this is discussed in the following subsection.

5.3.2 Implementation

The individual welding equipment parameters are programmed in the Microsoft Windows environment. Each parameter is programmed separately in terms of contiguous time segments. The user specifies the beginning and the end of each time segment, as well as the starting parameter value and the final parameter value for the given segment. If the starting value differs from the final value, the parameter ramps linearly between these two values during the segment duration. Obviously, the parameter is constant if the starting and the final values are the same. The user has the option of selecting pulsing of the parameter during the programmed segment, or not. The values of the "primary" and the "background" parameters are specified by the user, as well as their respective durations. This permits the programming of arbitrary pulsing frequency, duty cycle, peak value and background value. One segment can be specified as the "welding segment". This segment differs from the others in that its duration can be "open ended", i.e., it has a starting time but not a specified ending time. This allows the user to program a

welding segment for an indefinite duration, which is terminated by an external signal (e.g., from a pushbutton) or by pressing a key on the controller keyboard.

Figure 45 shows a photograph of the graphical screen, displaying typical programmed weld parameters. The uppermost window shows the *current*, ramped up to 150 Amperes in 5 seconds, and maintained at 150 Amperes for 20 seconds. Then the current is ramped down to 20 Amperes in a time period of 4 seconds, and from then to zero in additional 2 seconds. Below the arc current window are the *arc voltage* and *arc length* windows, side by side. In this example the arc length has been programmed to increase up to 0.100 in., as the current ramps up to the welding value. The arc length is maintained constant until the current is ramped down. The length ramps down from 0.100 in. as the current is reduced from 150 Amperes to 20 Amperes, and from then on the arc length is unchanged. The arc voltage will vary as shown in the arc voltage window. The arc voltage is calculated based on the varying arc current and arc length during the weld sequence. Finally, the programmed welding travel speed and the wire feed rate are shown at the bottom of the screen. Similar parameters are illustrated in Figure 46, which is printed directly to a printer from the Windows operating system. In this example the arc current is pulsed and, furthermore, the duration of the welding segment is indefinite, as indicated on the time scale by the label "T" at the end of the welding segment.

The weld parameter programming capabilities aid in controlling the heat input to the weld, and particularly in the tailout region. Extensive studies have been conducted on the problems of both Heat Affected Zone (HAZ) and Tailout or Crater Cracking. Special emphasis was placed on the use of pulsed welding process to achieve the desired improvement in cracking response. This method was also chosen because the Space Shuttle Main Engine (SSME) is currently being welded using pulsed GTAW.

A survey of the literature (Becker and Adams, 1979 [39]; Omar and Lundin, 1979 [40]; Thompson, et al., 1985 [41]; Burgardt and Heiple, 1986 [42]; Gordine, 1971 [43]; Savage and Aronson, 1966 [44]; D'Annessa, 1967 [45]; Savage, et al., 1965 [46]; Duvall and Owczarski, 1967 [47]) indicates two partially contradictory objectives when attempting to control cracking, particularly in Inconel 718. An increase in heat input tends to increase grain size in the HAZ, leading to an increase in HAZ microfissuring. However, since solidification growth morphology in the weld pool itself is directly affected in a beneficial manner by an increase in the temperature gradient, a means of providing a localized increase in the temperature gradient without a major increase in heat applied to the HAZ would be useful for decreasing the overall cracking tendency.

Complete control of the welding parameters is extremely important if weld quality is to be controlled. Particularly, it is important to maintain all control mechanisms active during the current rampdown period, just before the weld is termi-

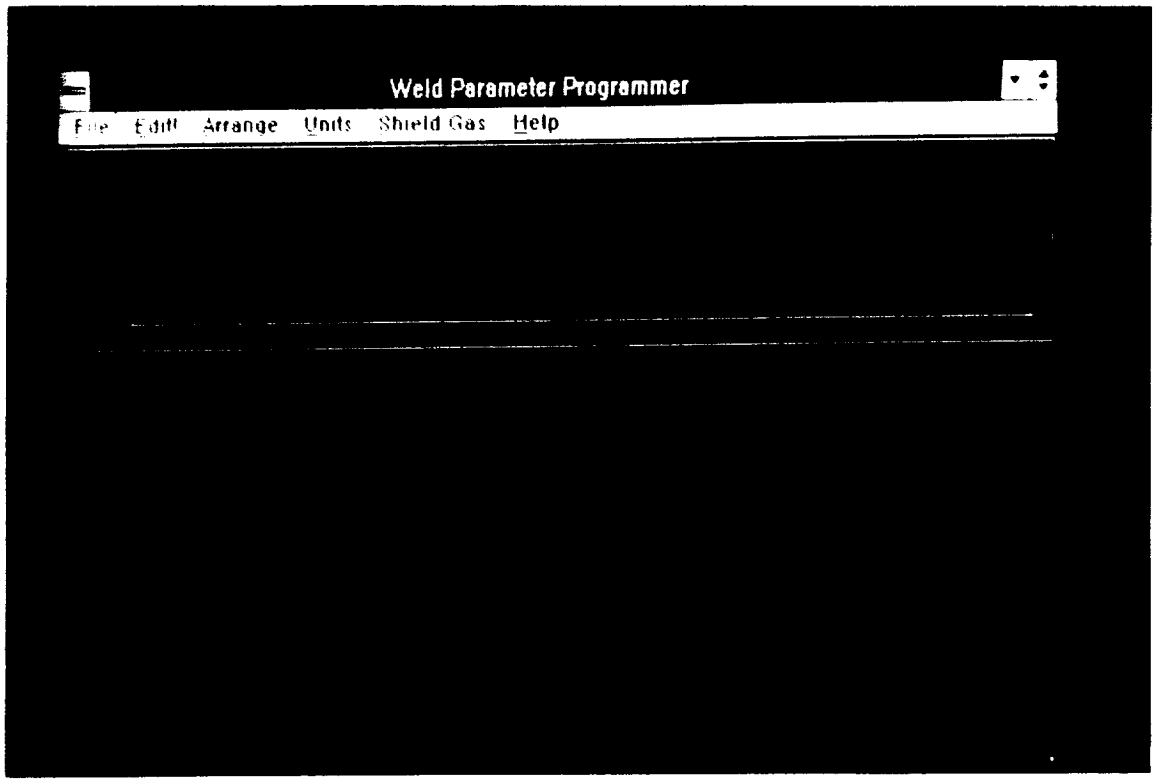


Figure 45. A photograph of the graphical weld sequence generator screen.

ORIGINAL PAGE IS
OF POOR QUALITY

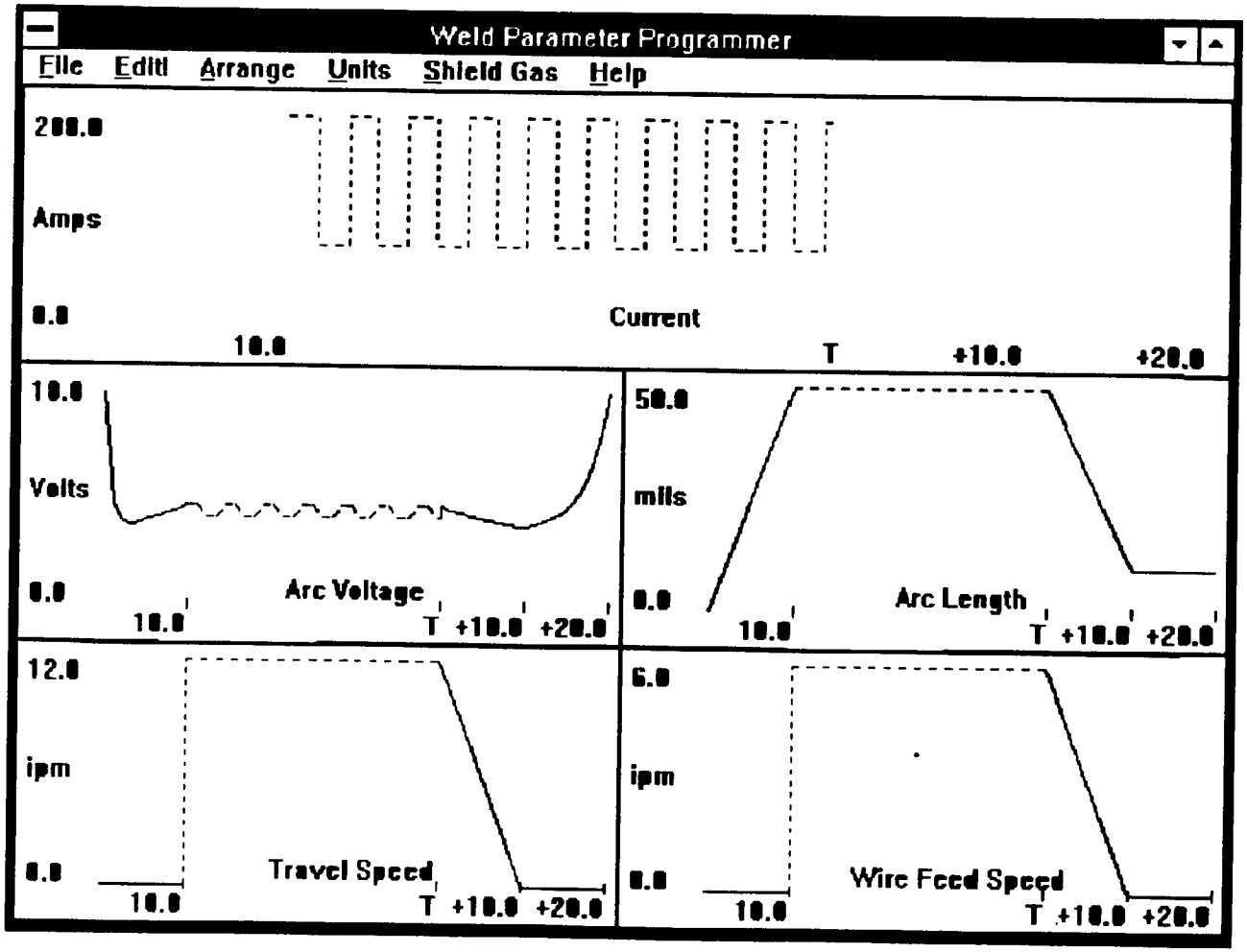


Figure 46. A printout of the welding parameters, as displayed in the Weld Parameter Programmer window.

nated. Investigations were performed to examine parameter control for reducing or eliminating the tendency of Inconel-718 to crack at the termination of the weld bead. The work published by Savage, Lundin, and Aronson, 1965; Rienks and Ashauer, 1971 []; and Omar and Lundin, 1979; indicate the following to be potentially beneficial in the control of crater cracking:

1. Control of pool shape - as a means of controlling grain orientation.

2. Control of temperature gradient - as a means of controlling grain type (cellular vs. dendritic).

3. Control of temperature gradient - as a means of controlling grain size.

Furthermore, Burgardt and Heiple, 1986, indicate that significant weld pool control can be accomplished by variation of the energy density of the welding arc. They state: "It is suggested that input power density is the fundamental quantity that describes the role of all welding variables in determining weld shape." This suggestion, as well as supporting results from a number of other researchers prompted the continuation of construction of a prototype AVC which can be maintained intact during the tail-out period as well as during the actual welding period.

From the physics standpoint the two variables most easily controlled were (2) temperature gradient and (3) heat addition. These could be changed by controlling the current, e.g. by ramping or by pulsing. Pulsing was chosen for an experiment since (a) the technique of pulsing has been determined to be a viable method for GTAW, (b) variation of the pool shape in the appropriate manner (from elongated to rounded) would require the synchronization of changes in input power and travel speed.

Examples of test results are shown in Figures 47, 48, and 49. Figure 47 (a) shows the results of an abrupt termination or step decrease in welding current. Figure 47 (b) shows the same weld after the application of dye penetrant crack detection solution. A crack is indicated by the arrow. Figure 48 (a) shows the weld resulting from a ramp or gradual decrease of welding current. The smaller crack due to this method is shown in Figure 48 (b).

The weld shown in Figure 49 (a) was terminated by use of a 10 Hz pulse rate with a duty cycle of 50%. The adjacent pulses were decreased by about 15% of the DC current value an thus this termination procedure was accomplished with about 6 pulses. Close examination of the termination region shows the gradually decreasing pool size as the termination process advances. Testing of this weld (and a number of others of terminated in the same manner) with crack detection solution showed no cracking (see Figure 49 (b)).

Control of temperature gradient and heat addition has thus been shown to be a feasible method for controlling cracking in the termination of weld beads in Inconel-718. However, the control of pool shape by travel speed control is a further means of control which can be implemented if cracking in a particular application is not eliminated by control of temperature gradient and heat input only.

Experiments to determine the potential for weld bead control using pulsed current (particularly relatively high energy density pulses) were performed. High energy density is desirable to provide adequate temperature gradient and penetration even though the heat input may be relatively small. The weld arc was examined photographically to determine arc energy density, paralleling the work of Saedi and Unkel, 1988, who used a photodiode array. These experiments showed that the radiation intensity and current density were positionally equivalent over a wide range of currents and arc lengths.

Development of the arc length control feature of the digital AVC allows the ramping of current downward during tailout while maintaining sufficient power density to obtain desired pool characteristics. This is accomplished by decreasing the arc length so that a well defined arc (hence weld pool) is maintained during the tailout process.

A comparison of the arc behavior with constant arc length and current rampdown (as used in conventional AVC systems) with arc behavior with both current and arc length rampdown (with enhanced AVC) is shown in Figures 50 and 51, respectively. In both Figures the current is decreased from right (maximum) to left (minimum). These Figures show a negative image of the arc (black) for clarity of reproduction. Note that in Figure 50, (constant arc length - current rampdown) that at the lowest current setting (left image), the arc has lost most of its structure and that there is no longer a molten pool. At this point the arc is serving only as a relatively diffuse heat source to adversely affect grain type and size.

However, Figure 51 (arc length proportional to current) shows that with a proportional decrease of arc length and current the geometrical structure of the arc is maintained for the duration of the tailout process. Even at the lowest current setting (50 Amperes in this case) a small but well defined weld pool is still present. This achieves the desired goals of maintaining pool shape while also maximizing temperature gradient with its resulting enhancement of desirable grain characteristics.

The digital weld controller allows the user to select either the constant arc length - high power density pulse mode, or the proportional arc length - arc current mode for tailout control.

98 — 100
PRECEDING PAGE [redacted] NOT FILMED
MISSING

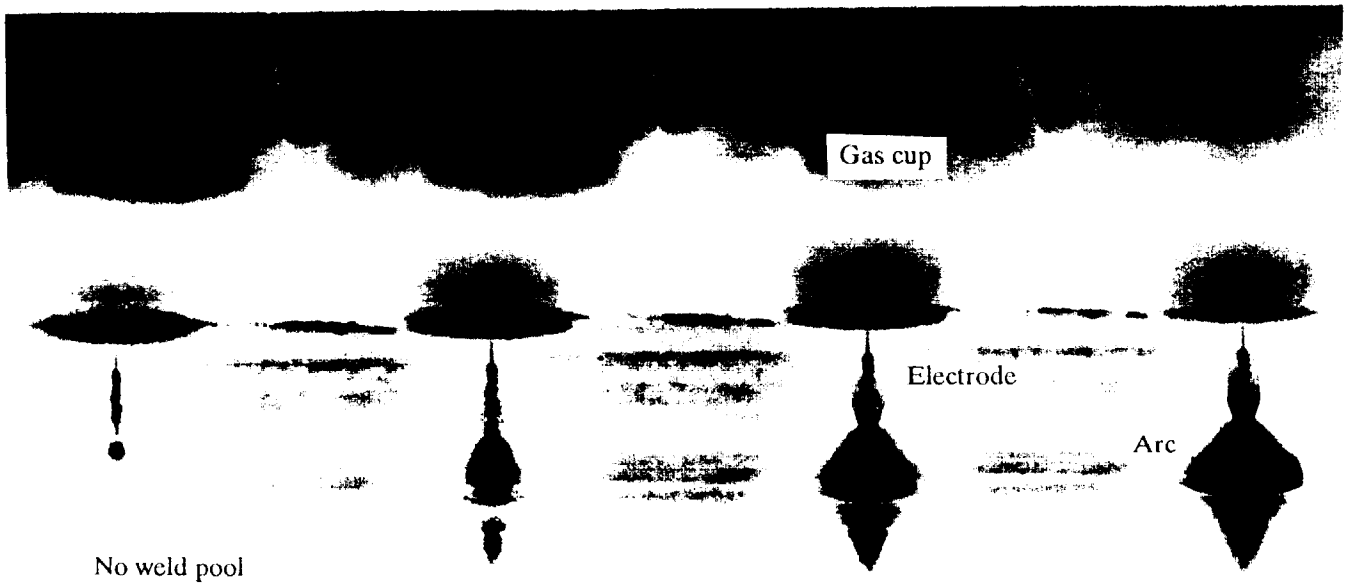


Figure 50. Illustration of conventional arc termination, with the arc length maintained fixed while the current is tailed out. Note that the weld pool disappears abruptly before the current reaches its minimum.

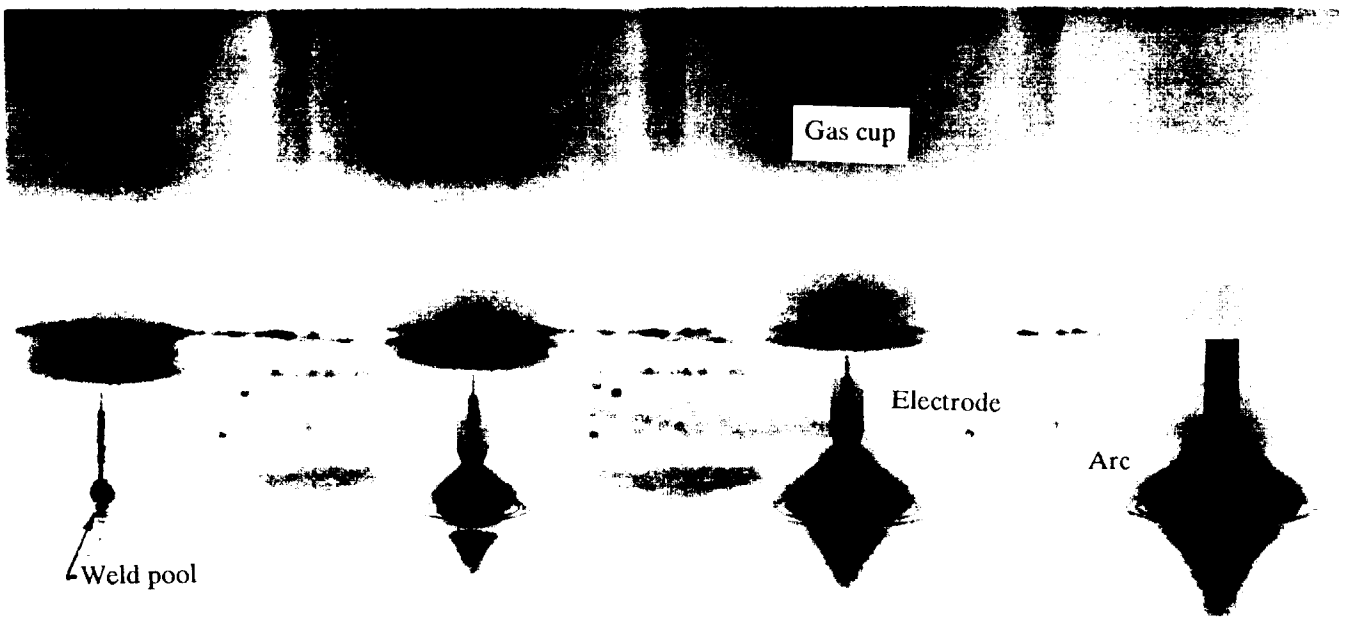


Figure 51. Illustration of an enhanced arc termination, with the arc length ramped down as the current is ramped out, as well. A well defined pool is maintained down to the lowest current levels.

5.4 Real-Time Control Loop Implementation

A real-time control loop was programmed for the 80386 computer on the system VME-bus. This control loop was implemented using the INT8 clock interrupt of the system.

The clock interrupt occurs approximately 18.7 times a second while the computer is running. This clock interrupt, which is programmed into the system Basic Input Output System (BIOS), has two purposes in the IBM-PC architecture: (i) to maintain track of the system time and date, and (ii) to control time delays used in the floppy disk controller.

A program was written to change this interrupt process in two respects. First, the counter register, which defines the interrupt frequency, is accessed so that this frequency can be altered. Second, the default interrupt service routine is intercepted and the tasks required for welding control are inserted into the interrupt. These welding control tasks include acquisition of sampled analog-digital channels, calculations of output signals, and placement of the output values into the digital-analog output ports. The frequency of this control loop is programmable, but it is ultimately limited by the complexity of calculations required for each cycle. In its present configuration, the maximum control loop update frequency is about 500 Hz. Because this frequency is more than adequate for the control tasks of interest here, such as AVC operation, current, travel speed, and wire feed rate control, the operations performed during each clock interrupt have not been optimized for speed at this time. If the need for faster control loop frequency arises, such optimizations can be carried out, but possibly at the expense of code readability. At this stage such undertakings are not considered worthwhile.

6. DATA RECORDING

The system is equipped with two distinct means for recording data.

A database is provided for entering welding records. The database stores information on each weld in a record. Standard fields have been created for the records, where the user can enter the various welding parameters, and various descriptions and comments. The database is implemented in the Windows environment and the user interface has been designed to be as intuitive as possible.

In addition to the database, which usually is accessed off-line, the system is provided with a dedicated data acquisition system for recording the welding parameters in real-time. Through real-time recording of the welding parameters, detailed and accurate information on the welding sequence can be acquired and reviewed after welding.

6.1 Weld Database

A relatively simple database for the welding system, constructed using the dBase IV database program, was implemented early in the work period. That database version was submitted to NASA with one of the quarterly progress reports, for evaluation. In addition to serving as a welding database, it provided a menu-oriented user interface for Nunes' model. After considerable experimentation with this system it was decided to abandon this implementation. The reasons for this were primarily two. First, the dBase IV system, version 1.0 (which was the latest version at that time) proved to very memory intensive and slow in execution. Its memory requirements were so hefty that it would marginally run on standard MS-DOS based computers. Expanded memory was frequently required if the operating system and other memory resident programs took up a relatively moderate space in the computer. The sluggishness of the dBase IV program was another annoying aspect. Simple switching between various parameter menus for the weld model, e.g., switching between entering model quadrupoles and modifying the base metal specifications, frequently took a few seconds. Furthermore, no version of the dBase program was available for the Microsoft Windows environment. This was a serious drawback once the Windows implementation had been decided.

Despite an extensive search, no suitable database programs were available for the Windows system at the time when decision about the database had to be made, although a few database systems are available for Windows as of this writing. Therefore it was decided to write a separate database program for the system. This program, as all other code written for this research, is developed with the Microsoft C compiler. A sample window of this database is shown in Figure 52. The database fields shown on this screen are those which were determined to be most important for database recording. The Name field holds the name of the person

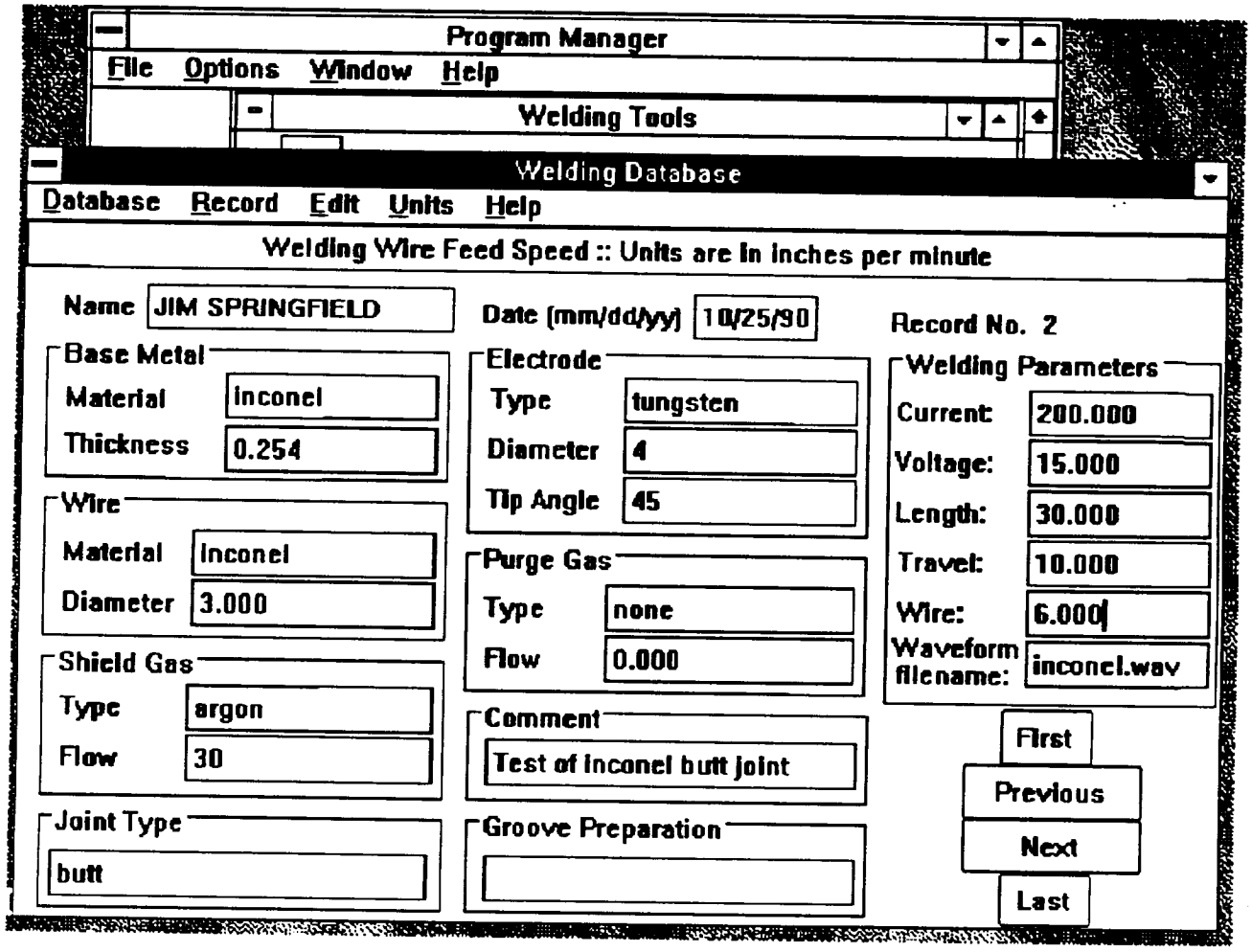


Figure 52. A sample database menu window.

who creates the record, and the date is the date when the record is created. The record number (2, in this case) is automatically assigned to the record when it is created, and the database system accesses records based on their numbers. The Base Metal subwindow contains the Material field, which names the base metal, and Thickness, which stores the thickness of the welded piece. The Wire subwindow specifies the wire material and the wire diameter. Similarly, the Shield Gas window stores the gas type and the gas flow, and the Purge Gas window stores the backpurge gas type and flow. The Electrode window specifies the electrode material, its diameter, and the included (total) angle of the electrode tip. At the bottom of the database window are fields for the Joint Type, Groove Preparation, and Comment. The rightmost column of the database window contains the values for the parameters which control the welding process, i.e., the arc current, voltage, arc length, travel speed, and wire feed rate. These values are essentially those that are used for the main welding segment, rather than upsloping or downsloping values. These values, as displayed in the database menu, are actually extracted from a corresponding waveform file, which determines the entire sequence of weld parameters. The waveform file in this case is "inconel.wav", as displayed in the corresponding menu field.

In the bottom-right corner of the screen a set of control buttons (First, Previous, Next, Last) is provided to facilitate browsing through the records of the database. Other database operations are selected from the options displayed at the top of the window (Database, Record, Edit, Units, Help). It should be noted that here, as well as in other windows of the system, the user can select metric units or English units for numeric parameters. Parameters are always entered in the units currently selected. However, the switching of units at any time results in all screen values being recalculated and updated on the screen in the new units.

6.2 Data Acquisition

A data acquisition system is provided with the welding controller. This is system is microprocessor-based and is designed to provide quality control for critical welding operations. With the appropriate welding sensors or transducers installed, the system can monitor a welding system for the following parameters:

Gas flow rate	[cubic feet /hour]
Elapsed weld time	[seconds]
Wire feed rate	[inches/minute]
Arc current	[Amperes]
Arc voltage	[Volts]
Travel speed	[inches/minute]
Heat input	[KJoules/inch]
Temperature	[sensor dependent]

If pulsed welding is used, the following parameters can be monitored as well:

Peak current	[Amperes]
Peak voltage	[Volts]
Background current	[Amperes]
Background voltage	[Volts]
Pulsing frequency	[Hertz]

Limits can be defined for each parameter, so that the system signals the operator or triggers a corrective action if a parameter value falls outside the specified range.

The data acquisition system records the welding parameters at a fixed rate of 100 samples/second. The actual sampling rate, however, is 25,000 samples/second, but the samples gathered during each recording period are filtered and processed internally in the data acquisition unit. The sampled data is made available from the acquisition system through an RS-232 serial port. This port is usually connected to a microprocessor, where the data is stored and processed for further analysis. In the delivered system the serial port is connected to an 80286-based computer board on the system VME-bus, which stores the data, displays it, and performs the statistical analysis discussed in the following subsection.

6.3 Statistical Data Analysis

A software package is provided with the data acquisition system which permits the user to display the sampled data in tabular or graphical form. Furthermore, this software can perform various statistical analyses on the welding data, such as averaging, determining drifting trends, etc.

The data screen of the software displays the following variables:

Elapsed time	[Seconds or milliseconds]
Gas flow rate	[cfh or liters/minute]
Travel speed	[ipm or mm/second]
Wire feed rate	[ipm or mm/second]
Voltage	[Volts]
Current	[Amperes]
Date	[date of data collection]
Time	[time of data collection]
Weld1-5	[averaged parameters for specified welds]
Mean	[average parameter value for all welds]
Range	[range from highest to lowest value]

These parameters can be analyzed and displayed in various ways. An Averaging chart is a full scale graph of the averages of a specified parameter. The Range chart

displays graphically the difference between the maximum and the minimum of a weld parameter. This feature is useful for depicting any significant changes in the spread of the sampled data. A Histogram show the distribution of values for a given parameter, i.e., it indicates the spread of a specified parameter, but also how frequently the parameter deviated to a given extent. The Sigma chart is a graphical representation of the standard deviations of each of the records in the data file. Finally, the Tolerance chart provides a graphical display of the minimum and maximum values of each record for the selected parameter. This aids the user in identifying any abnormal values of the parameter.

7. DELIVERED SYSTEM

The control computer to be delivered to NASA is shown from the front in Figure 53, with the keyboard drawer and the printer drawer open.

The delivered system is mounted in a standard 19" industrial grade rack cabinet (NEMA 12 rated). The cabinet is equipped with a metal-framed glass door at the front and a metallic back door, both of which are lined with sealing rubber to keep dust out of the cabinet. The cabinet is mounted on casters. The computing equipment are housed in a VME-based workstation cabinet. This cabinet contains an EGA-compatible computer monitor, a 5-1/4" floppy disk drive and a sealed front panel keypad, in addition to the VME-bus slots and the internal power supply. An optional outlet for a keyboard is provided on the front panel.

Directly above the workstation cabinet is the data acquisition system, which can be controlled from its front panel as well as the computer which it is connected to on the VME bus. The system keyboard is mounted on a drawer below the VME workstation. The keyboard is a standard IBM-AT compatible keyboard, with a built-in trackball and three trackball control buttons. The trackball serves conveniently as a computer mouse when the Windows environment is used. Below the keyboard is a second drawer, which stores the system manuals and documentation, and at the bottom is the printer drawer. A dot matrix printer is provided, with an automatic switching device, so that either of the two computer boards in the system can access the printer.

A rear view of the system is shown in Figures 54 and 55, with the various components indicated with labels. 120 VAC power is routed through an inlet close to the bottom of the rack cabinet. The power cable is detachable. A shielded 26-pin connector for input and output signals is mounted on the same back plate as the power connector. This connector is wired for a total of 12 distinct signal channels, i.e., 8 output signals and 4 input signals. All signal wiring utilizes shielded two-wire cables, with the shield conductors detachable from the system cabinet if desirable.

The VME bus of the workstation is accessible from the rear of the cabinet. The VME bus is equipped with two computer boards (one 80386 and one 80286-based). The 80386 computer board serves to run Microsoft Windows and all software associated with it. The 80286 computer is dedicated to serve the data acquisition system. In addition to the two computers, the VME bus is equipped with one 8-channel analog-digital converter and one 8-channel digital-analog converter. The keyboard, mounted on its drawer, can be seen in these Figures, stored inside the system cabinet. Access to the analog input and output signals is through ribbon cables. To facilitate flexibility, signal connection boxes were constructed, where the ports connected to the ribbon cables were made more accessible on RCA coaxial connectors. Figure 51 offers a close-up view of the signal connection boxes. Each coaxial

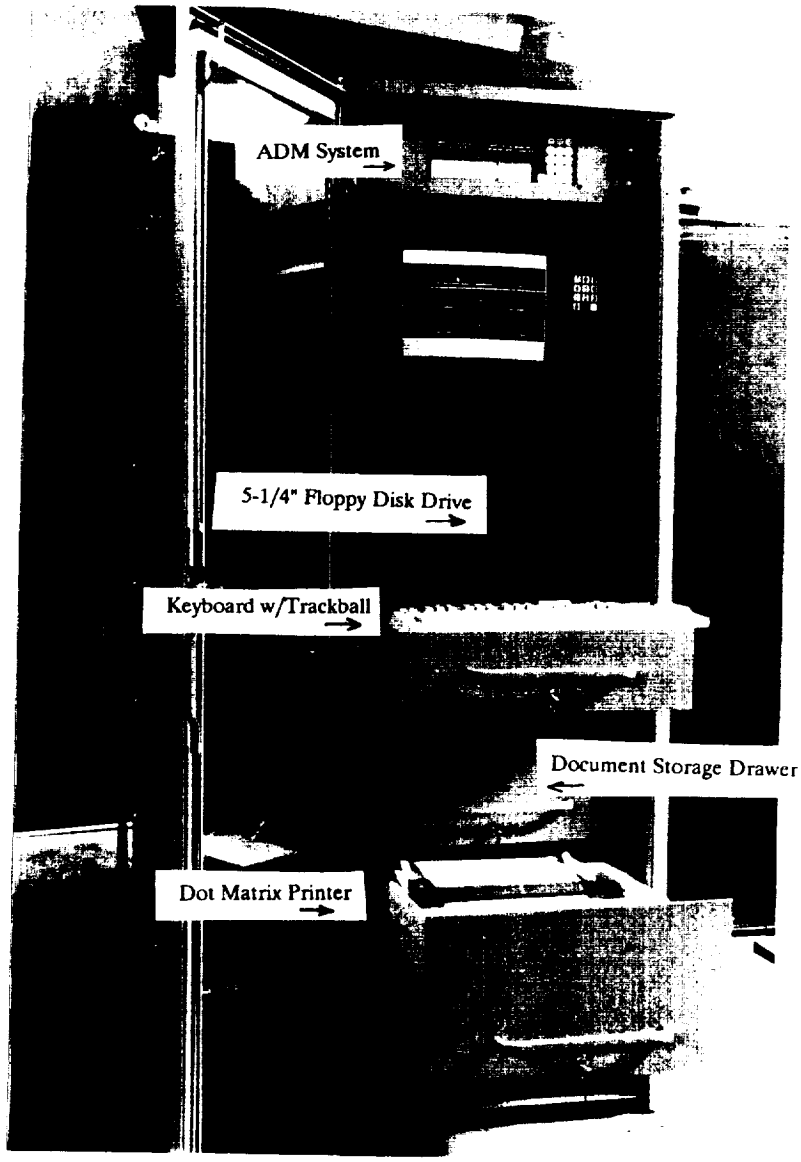


Figure 53. A front view of the welding controller.

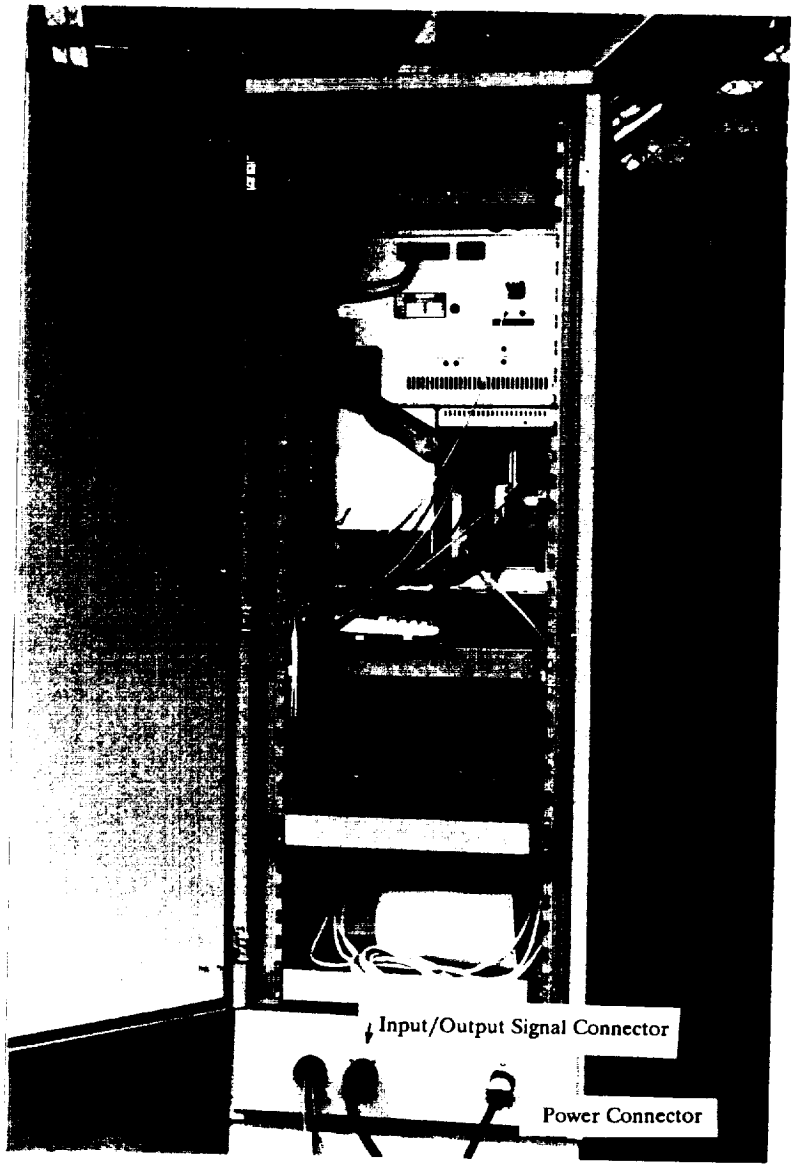


Figure 54. A rear view of the welding controller.

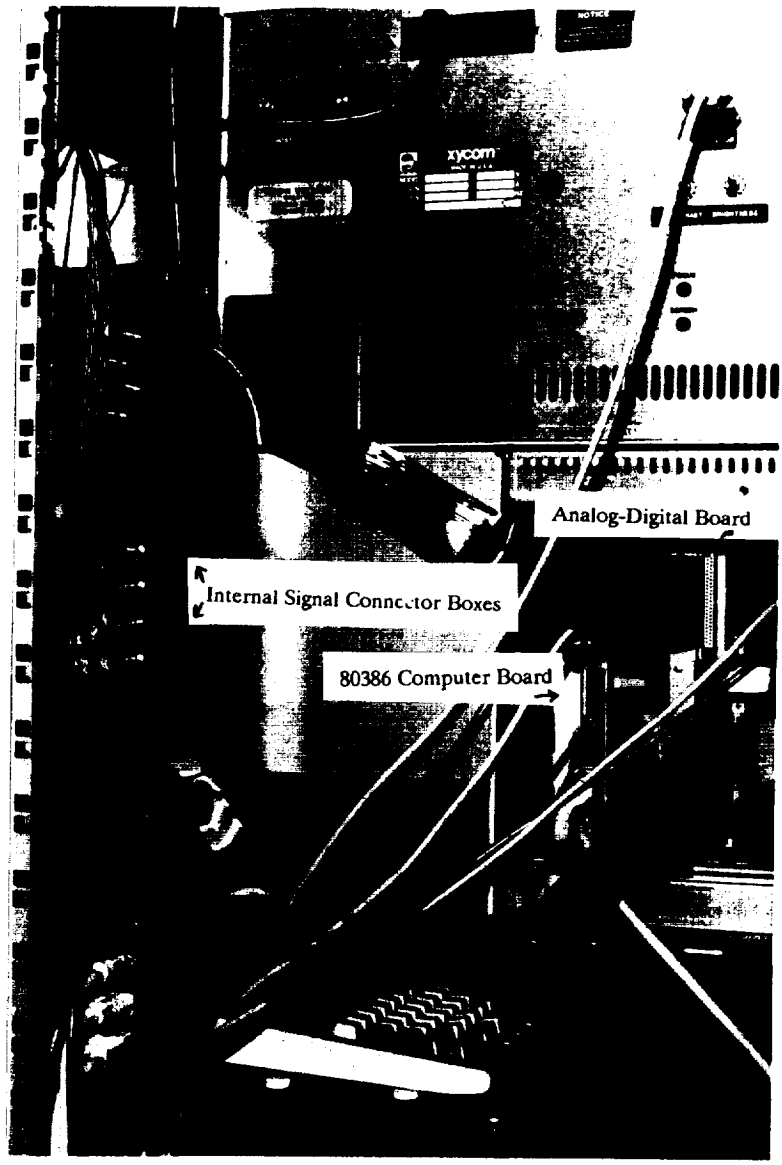


Figure 55. A close-up rear view of the welding controller.

connector is connected to a shielded cable which in turn is connected to the 26 pin signal connector.

All software is loaded on the internal hard disks of the system. Software and operating documentations, as well as system manuals are delivered with the system.

8. APPLICATION TO VPPAW AND OTHER TECHNOLOGIES

Although the system developed for this research was primarily designed for Gas Tungsten Arc Welding (GTAW), it can be adapted to other welding processes, such as Variable Polarity Plasma Arc Welding (VPPAW).

The welding controller hardware has been selected and assembled so as to be as modular as possible. The analog input and output signal lines are routed through signal connector boxes, mounted inside the system rack cabinet, and this permits easy access to all signals for testing, expansion, and rerouting of the individual signal channels.

The software is designed with modularity in mind, as well. The various software tools implemented under Windows, such as the database and the digital AVC, are essentially individual, self-supporting programs, that can be replaced with future versions, modified for other welding processes. This applies in particular to the weld modeling programs, which almost surely have to modified to some extent, if other processes are to be considered.

9. BIBLIOGRAPHY

1. Hunter, J.J., Bryce, G.W., Doherty, J., "On-line control of the arc welding process", Developments in Mechanised Automated and Robotic Welding, WI, 1980.

2. Smartt, H.B., Key, J.F., "An Investigation of Factors Controlling GTA Weld Bead Geometry", Trends in Welding Research in the U.S., EGG M10-581, Idaho, November 16-18, 1981.

3. Dornfeld, D.A., Tomizuka, M., Langari, G., "Modeling And Adaptive Control Of Arc Welding Processes", ASME Special Publication: Measurement and Control for Batch Manufacturing, November, 1982.

4. Hardt, D.E., Garlow, D.A., Weinert, J.B., "A Model of Full Penetration Arc welding for Control System Design", Transactions of the ASME, Journal of Dynamic Systems, Measurement, and Control, March 1985, pp. 40-46.

5. Andersen, K., Cook, G.E., Strauss, A.M., "A Generic Expert System for Materials Processing in Space", Proc. Conf. on Artificial Intelligence for Space Applications, pp. 227-236, MSFC, AL, NASA, Nov. 1986.

6. Shepard, M.E., Cook, G.E., and Randall, M.D., "Computer-Based Acquisition and Analysis of Arc Welding Signals", American Welding Society convention, Chicago, Illinois, April 1987.

7. Cook, G.E., Yizhang, L., and Shepard, M.E., "Computer-Based Analysis of Arc Welding Signals for Tracking and Process Control", IEEE 1986 Industrial Applications Society Convention, Oct. 1986.

8. Hayes-Roth, F., Waterman, D.A., Lenat, D.B., Building Expert Systems, Addison-Wesley, New York, 1983.

9. Sliwinski, K.E., Ruokangas, C.C., "Adaptive Robotic GTA Welding for the SSME: An Integrated System", Conference Proceedings, Robots-11, 17th International Symposium on Industrial Robots, Chicago, April 26-30, 1987.

10. Cook, G.E., "Robotic Arc Welding: Research in Sensory Feedback Control", IEEE Transaction on Industrial Electronics, vol. IE-30, No. 3, pp. 252-268, August 1983.

11. McCampbell, W.M., Cook, G.E., Nordholt, L.E., and Merrick, G.I., "Development of an Intelligent Weld System", Welding Journal, vol. 45, No. 3, pp. 139s-144s, March 1966.

12. Richardson, R.W., et al., "Coaxial Arc Weld Pool Viewing for Process Monitoring and Control", Welding Journal, pp. 43-50, March, 1984.

13. Maram, J.M., "Optical Weld Contour Monitor for Penetration Control", NASA Tech Briefs, p. 142, March/April, 1986.

14. Maram, J.M., Collins, J., "Acoustic Emission Weld Penetration Monitor",

NASA Technical Utilization Disclosure, July, 1985.

15. Salter, R.J. and Deam, R.T., "A Practical Front Face Penetration Control System for TIG Welding," *Developments in Automated and Robotic Welding*, Ed.: D.N. Waller, Cambridge, England, The Welding Institute, pp. 38-1 to 38-12 (1987).

16. Deam, R.T., "Weldpool Frequency: A New Way to Define a Weld Procedure," Paper Presentation at *2nd International Conference on Trends in Welding Research*, Gatlinburg, Tennessee. Co-chairman: S.A. David and J.M. Vitek, ASM International, May 14-18, 1989.

17. Chin, B.A. et al, "Infrared Thermography Shows Promise for Sensors in Robotic Welding", Robotics Today, February 1983.

18. Cook, G.E., "Feedback and Adaptive Control in Automated Arc Welding Systems," Metal Construction, vol. 13, No. 9, pp. 551-556, Sept. 1981.

19. Eraslan, A.H., Zacharia, T., Aidun, D.K., "WELDER: A Computer Code For Simulating Fast-Transient, Three-Dimensional, Three-Phase, Flow, Temperature And Material Composition Conditions During Welding", Report No. MIE-142, Dept. of MIE, Clarkson University, Potsdam, NY., October 1986.

20. Rosenthal, D., "Mathematical Theory of Heat Distribution During Welding and Cutting", Welding Journal, 20(5), 1941, pp. 220s-234s.

21. Nunes, A.C., "An Extended Rosenthal Weld Model", Welding Journal, June 1983, pp. 165s-170s.

22. Mills, G.S. "Fundamental Mechanisms of Penetration in GTA Welding", Welding Journal, 58(1), 1979, pp. 21s-24s.

23. Tsai, N., "Heat Distribution and Weld Bead Geometry in Arc Welding", Ph.D. Thesis, Department of Materials Science, M.I.T., April 1983.

24. Smartt, H.B., Stewart, J.A., Einerson, C.J., "Heat Transfer In Gas Tungsten Arc Welding", ASME Metals/Materials Technology Series, 1985 ASM International Welding Congress, October 1985.

25. McCulloch, C.A., Pitts, W., "A Logical Calculus of the Ideas Imminent in Nervous Activity", Bulletin of Mathematical Biophysics, 5, pp. 115-133, 1943.

26. Widrow, B,. Hoff, M.E., "Adaptive Switching Circuits", 1960 IRE WESCON Conv. Record, Part 4, pp. 96-104, August 1960.

27. Hopfield, J.J, "Neural Networks and Physical Systems with Emergent Collective Computational Abilities", Proc. National Academy of Sciences, USA, vol. 79, pp. 2554-2558, April 1982.

28. Lippmann, R.P., Gold, B., Malpass, M.L., "A comparison of Hamming and Hopfield Neural Nets for Pattern Classification", MIT Lincoln Laboratory Technical Report, TR-769.

29. Rumelhart, D.E., Hinton, G.E., Williams, R.J., Parallel Distributed Process-

ing, volumes 1-2, MIT Press, Cambridge, MA, 1986.

30. Lippmann, R.P., "An Introduction to Computing with Neural Nets", IEEE ASSP Magazine, pp. 4-22, April 1987.

31. Savage, W.F., Nippes, E.F., and Zanner, F.J., "Determination of GTA Weld-Puddle Configurations by Impulse Decanting", *Welding Journal, Welding Research Supplement*, Vol. 57, No. 7, pp. 201s-210s, July 1978.

32. Cook, G.E., Andersen, K., Barnett, R.J., "Feedback and Adaptive Control in Welding" (Invited Keynote Paper), *Recent Trends in Welding Science and Technology, 1989: Proc. 2nd International Conference on Trends in Welding Research*, eds. S.A. David and J.M. Vitek, American Society for Metals (ASM) International, Gatlinburg, Tennessee, May 14-18, 1989, pp. 891-903.

33. Thompson, E.G., "Heat Cracking of Inconel-718 in Zones Affected by Weld Heat", *Metallurgy, A Compilation*, NASA Technology Utilization, NASA SP-5940 (01)

34. Gordine, J., "Some Problems in Welding Inconel 718", *Welding Journal, Welding Research Supplement*, Vol. 50, No. 11, pp. 480s-484s, November 1971.

35. Mills, W.J., "Effect of Heat Treatment on the Tensile and Fracture Toughness Behavior of Alloy 718 Weldments", *Welding Journal, Welding Research Supplement*, Vol. 63, No. 8, pp. 237s-245s, August 1984.

36. Savage, W.F. and Lundin, C.D., "The Varestraint Test", *Welding Journal, Welding Research Supplement*, Vol. 44, No. 10, pp. 433s-442s, October 1965.

37. Ogata, K., *Modern Control Engineering*, Prentice-Hall, Inc., N.J. 1970.

38. Becker, D.W., and Adams, C.M., "The Role of Pulsed GTA Welding Variables in Solidification and Grain Refinement", *Welding Journal, Welding Research Supplement*, Vol. 58, No. 5, pp. 143s-152s, May 1979.

39. Omar, A.A., and Lundin, C.D., "Pulsed Plasma - Pulsed GTA Arc: A Study of the Process Variables', *Welding Journal, Welding Research Supplement*, Vol. 58, No. 4, pp. 97s-105s, April 1979.

40. Thompson, R.G., Cassimus, J.J., Mayo, D.E., and Dobbs, J.R., "The Relationship Between Grain Size and Microfissuring in Alloy 718", *Welding Research Supplement*, Vol. 64, No. 4, pp. 91s-96s, April 1985.

41. Burgardt, P., and Heiple, C.R., "Interaction between Impurities and Welding Variables in Determining GTA Weld Shape", *Welding Research Supplement*, Vol. 65, No. 6, pp. 150s-155s, June 1986.

42. Savage, W.F., and Aronson, A.H., "Preferred Orientation in the Weld Fusion Zone", *Welding Journal, Welding Research Supplement*, Vol. 45, No. 2, pp. 85s-89s, February 1966.

43. D'Annessa, A.T., "Microstructural Aspects of Weld Solidification", *Welding*

Journal, Welding Research Supplement, Vol. 46, No. 11, pp. 491s-499s, November 1967.

44. Savage, W.F., Lundin, C.D., and Aronson, A.H., "Weld Metal Solidification Mechanics", *Welding Journal, Welding Research Supplement*, Vol. 44, No. 4, pp. 175s-181s, April 1965.

45. Duvall, D.S., and Owczarski, W.A., "Further Heat-Affected-Zone Studies in Heat-Resistant Nickel Alloys", *Welding Journal, Welding Research Supplement*, Vol. 46, No. 9, pp. 423s-432s, February 1967.

46. Rienks, F., and Ashauer, R.C., "Development and Evaluation of a Modulated Power Control for Fusion Welding", *Welding Journal, Welding Research Supplement*, Vol. 50, No. 5, pp. 222s-230s, May 1971.

47. Goodwin, G.M.,. "The Effects of Heat Input and Weld Process on Hot Cracking in Stainless Steel", Welding Journal, Welding Research Supplement, Vol. 67, No. 4, pp. 88s-94s, April, 1988.

Appendix A: System Hardware Specifications

XYCOM ASM XVME 683 Microprocessor Board

Processor:	80386
Processor Clock Frequency:	25 MHz
Math Coprocessor:	80387
Graphics Controller:	VGA/EGA/MDA/Hercules
Floppy Disk:	5-1/4", 1.2MB/360KB, SA450 Interface
Hard Disk:	160MB Conner, SCSI Interface
On-Board Memory:	4MB Dual-Port RAM
Serial Ports (2):	RS-232
Parallel Port (1):	Centronics Compatible
Bus Type:	VME
Real Time Clock.	

Manufacturer: XYCOM, Inc.
750 North Maple Road
Saline, MI 48176
(313) 429-4971

XYCOM ASM XVME 682 Microprocessor Board

Processor:	80286
Processor Clock Frequency:	10 MHz
Math Coprocessor:	80287 (not included)
Graphics Controller:	EGA/CGA/MDA/Hercules
Floppy Disk:	5-1/4", 1.2MB/360KB, SA450 Interface
Hard Disk:	40MB, ST506
On-Board Memory:	2MB Dual-Port RAM
Serial Ports (2):	RS-232
Parallel Port (1):	Centronics Compatible
Bus Type:	VME
Real Time Clock.	

Manufacturer: XYCOM, Inc.
750 North Maple Road
Saline, MI 48176
(313) 429-4971

XYCOM ASM XVME 500/3 Analog-Digital Converter Board

Number of Channels:	16 Single Ended, 8 Differential
Resolution:	12 bits
Analog Input Full Scale Voltage, Unipolar:	0..10V
Analog Input Full Scale Voltage, Bipolar:	+/-5V, +/-10V
Linearity:	+/- 0.5 LSB
Common Mode Rejection Ratio:	> 60 dB
Conversion Time:	10 microseconds
Settling Time:	10 microseconds
Throughput Frequency:	50 KHz
System Accuracy, @ Gain = 1:	+/- 0.01% FSR
System Accuracy, @ Gain = 10,100:	+/- 0.1% FSR
System Accuracy Temperature Drift, @ Gain = 1:	40ppm/C
System Accuracy Temperature Drift, @ Gain = 10:	75ppm/C
System Accuracy Temperature Drift, @ Gain = 100:	110ppm/C

Manufacturer: XYCOM, Inc.
750 North Maple Road
Saline, MI 48176
(313) 429-4971

XYCOM ASM XVME 530 Digital-Analog Converter Board

Number of Analog Channels:	8
Resolution:	12 bits
Analog Output:	0..5V, 0..10V, +/-2.5V, +/-5V, +/- 10V
Linearity:	+/- 0.5 LSB
Common Mode Rejection Ratio:	> 60 dB
Conversion Time:	42 microseconds
Settling Time:	8 microseconds
Accuracy:	+/-0.5% LSB, +/-0.0122%

Manufacturer: XYCOM, Inc.
750 North Maple Road
Saline, MI 48176
(313) 429-4971

XYCOM 4251 VME-Bus Industrial Workstation

Bus Type:	VME
Monitor:	EGA Compatible

Manufacturer: XYCOM, Inc.
750 North Maple Road
Saline, MI 48176
(313) 429-4971

ADM III Arc Data Monitor

Input Sensors:

- Up to 3 Gas Flow Transducers
- Current/Voltage Sensor
- Wire Speed Transducer
- Travel Speed Transducer
- Temperature Sensor

Output Signals:

- Four External Alarm Relays
- LED Fault Status Lights
- Two-Line 40-Character Display
- RS-232 Serial Port
- Centronics-Compatible Parallel Port

Manufacturer: CRC-Evans Automatic Welding
11601 North Houston Road
Houston, TX 77086
(713) 999-8920

Epson LQ-510 Dot Matrix Printer

Vendor: Epson America, Inc.
2780 Lomita Boulevard
Torrance CA 90505

Chicony KB-5581 Keyboard with Trackball

Tecnorack Rack Cabinet

- Sealed Enclosure (NEMA 12, IP 55)
- Standard 19" Rack Width and Mounting Holes
- Keyboard Drawer
- Printer Drawer
- Auxiliary Drawer
- Casters

Manufacturer: Schroff, Inc.
170 Commerce Drive750 North Maple Road
Warwick, RI 02886
(401) 738-7988

Additional hardware components include switching devices, cables, connectors, etc.

Appendix B: Model Source Code

FILE CROSSECT.C

```
/*      -- File CROSSECT.C

        Author: Kristinn Andersen
        Date:   August 8, 1990.

        Revisions:
                Kristinn Andersen, August 28, 1990.

        There are three primary functions in this file:

        frontsection()  - finds a front view cross section of the weld pool;
        sidesection()   - finds a side view cross section of the weld pool;
        topsection()    - finds a top view cross section of the weld pool.

        Furthermore, two auxiliary functions are defined in this file:

        length()        - returns the length of given coordinates;
        extrapolate()   - returns new coordinates, extrapolated from
                          a pair of given coordinates.

        Each of the 3 primary functions has 5 arguments.
          1: The location of the cross sectioning plane.
          2: The spacing between the points along the traced boundary.
        3,4: Pointers to arrays that contain the found coordinate points
             of the boundary.
          5: A pointer to an integer number, specifying the number of
             elements returned in the boundary point arrays.

        Each function finds a starting boundary point by using the function
        linsrch() (defined in file SRCHMELT.C).  Then repeated calls to
        function angsrch() (defined in file SRCHMELT.C) are used to find
        successive boundary points, until the entire boundary has been
        traced.  The function angsrch() searches for a new boundary point by
        searching along a chord at a given radius from a previous boundary
        point.

        If a pool boundary was successfully found, each of the three
        functions returns 0.
        Otherwise any of the function may return any of the following:

        1: The pool was "too small" or not melting at all took place;
        2: The pool "too large", i.e. the number of traced points exceeded
           the constant MAXPTS.

        Note the definition of coordinate axes:

                X-axis:  Along torch travel direction;
```

```
                Y-axis:  Points to the "right", seen from torch;
                Z-axis:  Directed into the weld pool, normal to the plate.

                Origin is at the plate surface, in the center of the arc.
 */

 #include <malloc.h>
 #include <math.h>
 #include <stdio.h>
 #include "wmodel.h"

 /* #define PRINTPOINTS 1 */
 #define MAXPTS 1000

 enum cartesian {x_axis, y_axis, z_axis};

 double
   length(struct coordinates coord);
 struct coordinates extrapolate
   (struct coordinates coord1, struct coordinates coord2);
 struct coordinates linsrch_melt
   (enum cartesian axis, struct coordinates coord);
 struct coordinates angsrch_melt
   (enum cartesian axis, struct coordinates coord1, struct coordinates coord2);
 int
   frontsection(double x, double stepsize, double *y, double *z, int *n);
 int
   sidesection(double y, double stepsize, double *x, double *z, int *n);
 int
   topsection(double z, double stepsize, double *x, double *y, int *n);

 double
   length(struct coordinates coord)
 {
   return
     (sqrt(coord.x*coord.x+coord.y*coord.y+coord.z*coord.z));
 };

 struct coordinates extrapolate
   (struct coordinates coord1, struct coordinates coord2)
 {
   struct coordinates new_coord;

   new_coord.x = 2.0*coord2.x-coord1.x;
   new_coord.y = 2.0*coord2.y-coord1.y;
   new_coord.z = 2.0*coord2.z-coord1.z;
   return (new_coord);
 };

 int frontsection(double x, double stepsize, double *y, double *z, int *n)
 /* ------------------------------------------------------------------
```

```
   Finds a front view cross section of the weld pool (i.e., a section in
   the plane normal to the x-axis). The torch is moving towards the viewpoint.

   x:          Displacement of the cross section plane.
   stepsize:   Desired distance between the points along the pool boundary.
   *y, *z:     Pointers to two vectors, one containing the y-axis values
               and the other holding the z-axis values of the boundary
               coordinates, respectively.  These vectors must be allocated
               before this function is called.
   *n:         Number of found boundary points, returned from the function.
   ------------------------------------------------------------------- */

{
  struct coordinates
    coord1 = {0.0, 0.0, 0.0},
    coord2 = {0.0, 0.0, 0.0},
    last_coord2;
  int i;
  double temp;

  /* Initial point found on the plate surface, on right half of the pool
     boundary (with respect to the traveling torch).
     Global variable "Bead.radius" used in a first guess.
  */
  coord1.y = sqrt(Bead.radius*Bead.radius); /* NOTE: assumes x=0 */
  coord1.x = x;
  coord1 = linsrch_melt (y_axis, coord1);
  if (!length(coord1))
    return(1);

  y[0] = coord1.y;      /* First point stored as y and z coordinates. */
  z[0] = coord1.z;
  *n = 1;

#ifdef PRINTPOINTS
  printf("\n\n=== Frontsection points ===\n\n i:  y[i]:  z[i]:");
  printf("\n%3d %6.3lf %6.3lf", *n-1, y[*n-1], z[*n-1]);
#endif

  Bead.radius = length(coord1); /* Update radius and dT/dr with new info. */
  Bead.dTdr = -V*I*ETA/(1000.0*2.0*PI*K*Bead.radius*Bead.radius);
  coord2 = coord1;
  coord2.z = stepsize;
  coord2 = angsrch_melt(x_axis, coord1, coord2);
  if (!length(coord2))
    return(1);

 while (coord2.y>=0.0)  /* Find successive boundary points. */
 {
   if (coord2.z<THICKNESS)
   {
```

```
  y[*n] = coord2.y;
  z[*n] = coord2.z;
  (*n)++;
  if (*n>MAXPTS)
    return(2);
  last_coord2 = coord2;
  coord2 = extrapolate(coord1, coord2);
  coord1 = last_coord2;
  coord2 = angsrch_melt(x_axis, coord1, coord2);
  if (!length(coord2))
    return(1);
}
else
{
  if (z[*n-1]<THICKNESS)
  {
    y[*n] =  y[*n-1]+(THICKNESS-z[*n-1])*(y[*n-2]-y[*n-1])/(z[*n-2]-z[*n-1]);
    z[*n] = THICKNESS;
  }
  else if (z[*n-1]==THICKNESS)
  {
    y[*n] = y[(*n)-1]-stepsize;
    z[*n] = THICKNESS;
  }
  (*n)++;
  if (*n>MAXPTS)
    return(2);

  if ((temp=(*Model.temp_model)(x,y[(*n)-1]-stepsize,THICKNESS))>T_MELT)
  {
    /*  Next point will also be at the plate bottom. */
    coord2.x = x;
    coord2.y = y[(*n)-1]-stepsize;
    coord2.z = THICKNESS;
  }
  else
  {
    /* Next point will be the first one above the plate bottom (i.e.,
       within the plate) after one or more boundary points were at
       the bottom.
       The last plate-bottom point is coord1, and an initial guess
       for coord2 is at 45 degrees elevation from the plate bottom.
    */
    coord1.x = x;
    coord1.y = y[*n-1];
    coord1.z = z[*n-1];
    coord2.x = x;
    coord2.y = y[*n-1]-stepsize/sqrt(2.0);
    coord2.z = THICKNESS-stepsize/sqrt(2.0);
    coord2 = angsrch_melt(x_axis, coord1, coord2);
    if (!length(coord2))
```

```
            return(1);
        };
      };

#ifdef PRINTPOINTS
  printf("\n%3d %6.3lf %6.3lf", *n-1, y[*n-1], z[*n-1]);
#endif

  };
  y[*n] = 0.0;
  z[*n] = z[*n-1]+(z[*n-1]-coord2.z)*y[*n-1]/(coord2.y-y[*n-1]);
  (*n)++;
  if (*n>MAXPTS)
    return(2);

#ifdef PRINTPOINTS
  printf("\n%3d %6.3lf %6.3lf", *n-1, y[*n-1], z[*n-1]);
#endif

  for (i=*n; i<2*(*n)-1; i++)
  {
    y[i] = -y[2*(*n)-i-2];
    z[i] = z[2*(*n)-i-2];

#ifdef PRINTPOINTS
  printf("\n%3d %6.3lf %6.3lf", i, y[i], z[i]);
#endif

  };
  *n = 2*(*n)-1;
  return(0);
};


int sidesection(double y, double stepsize, double *x, double *z, int *n)
/* ------------------------------------------------------------------------
  Finds a side view cross section of the weld pool (i.e., a section in
  the plane normal to the y-axis).

  y:          Displacement of the cross section plane.
  stepsize:   Desired distance between the points along the pool boundary.
  *x, *z:     Pointers to two vectors, one containing the x-axis values
              and the other holding the z-axis values of the boundary
              coordinates, respectively.  These vectors must be allocated
              before this function is called.
  *n:         Number of found boundary points, returned from the function.
  ------------------------------------------------------------------------

  The function is basically structured as follows:

        Find 1st boundary point on plate surface, at pool front.
```

```
        Find new point in the plate, by searching around 1st point.

        WHILE new point is below plate surface:
           IF new point is not at plate bottom:
              record it and find (next) new point using angsrch();
              // Go back to WHILE with the new point //
           ELSE
              IF the point before new point was above plate bottom:
                 use interpolation to plate bottom and place new point there;
              ELSE IF the point before new point was at plate bottom:
                 step a fixed distance along plate bottom to place new point;
                 // Now, the new point has been placed at the plate bottom //

              IF melting occurs at the plate bottom, one step beyond the
                    new point that was just placed at the bottom:
                 set (next) new point to that coordinate, at plate bottom;
              ELSE
                 find (next) new point using angsrch();
              // Go back to WHILE with the new point //
        END
*/

{
  struct coordinates
    coord1 = {0.0, 0.0, 0.0},
    coord2 = {0.0, 0.0, 0.0},
    last_coord2;
  int i;
  double temp;

  /* Initial point found on the plate surface, on front half of the pool.
     Global variable "Bead.radius" used in a first guess.
  */
  coord1.x = sqrt(Bead.radius*Bead.radius); /* NOTE: assumes y=0 */
  coord1.y = y;
  coord1 = linsrch_melt (x_axis, coord1);
  if (!length(coord1))
    return(1);

  x[0] = coord1.x;      /* First point stored as x and z coordinates. */
  z[0] = coord1.z;
  *n = 1;

#ifdef PRINTPOINTS
  printf("\n\n=== Sidesection points ===\n\n i:  x[i]:  z[i]:");
  printf("\n%3d %6.3lf %6.3lf", *n-1, x[*n-1], z[*n-1]);
#endif

  Bead.radius = length(coord1); /* Update radius and dT/dr with new info. */
  Bead.dTdr = -V*I*ETA/(1000.0*2.0*PI*K*Bead.radius*Bead.radius);
  coord2 = coord1;
```

```
coord2.z = stepsize;  /* Second point assumed to be below first point. */
coord2 = angsrch_melt(y_axis, coord1, coord2);
if (!length(coord2))
  return(1);

/*  Find successive pool boundary points.  Start at the front of the
    weld boundary, at the upper surface of the plate, trace the boundary
    points first "down and backwards" under the pool, and continue until
    the surface is reached on the back side of the weld pool.
    The x-coordinate proceeds from a positive value ("front" of the pool)
    to a negative value.  The z-coordinate starts at 0 (upper surface
    of the plate), increases (into the welded plate), decreases again,
    and the search stops when z=0 again.  The y-coordinate is constant.
    Note that this does not reveal pools split up into disjoint segments.
    However, the function reveals disjoint areas of full penetration.
*/
while (coord2.z>=0.0)
{
  if (coord2.z<THICKNESS)     /* If next coord2 is inside plate */
  {
    x[*n] = coord2.x;
    z[*n] = coord2.z;
    (*n)++;
    if (*n>MAXPTS)
      return(2);
    last_coord2 = coord2;
    coord2 = extrapolate(coord1, coord2);
    coord1 = last_coord2;
    coord2 = angsrch_melt(y_axis, coord1, coord2);
    if (!length(coord2))
      return(1);

    /* New coord2 has been determined, and program continues back to the
       while-statement, above.
    */
  }
  else
  /* If next coord2 is at or below the bottom of the plate (i.e., full
     penetration has been reached), the pool tracing must be limited within
     the plate.  The coord2 is therefore replaced with a new, "truncated"
     coordinate, located at the bottom of the plate.  This replacement
     coordinate calculation depends on whether the truncated coordinate is
     the first one in a series of coordinates along the plate bottom, or if
     follows one or more coordinates at the bottom.
  */
  {
    if (z[*n-1]<THICKNESS)
    {
      /*  Next coord2 is the first one reaching to or beyond the plate
          bottom.  The next x[*n] is determined to be at the intersection
          between:
```

```
        the line connecting the last coord2 and the next coord2
        (i.e., the one that is at or beyond plate bottom),
      and
        the plate bottom.
      The z[*n] is simply set at the plate bottom.
   */
   x[*n] = x[*n-1] +
           (THICKNESS-z[*n-1])*(x[*n-1]-x[*n-2])/(z[*n-2]-z[*n-1]);
   z[*n] = THICKNESS;
 }
 else
 {
   /*  Last coord2 was already at the plate bottom.  The next coord2
       is at or beyond the plate bottom, and therefore it is set at
       the plate bottom, too.
   */
   x[*n] = x[(*n)-1]-stepsize;
   z[*n] = THICKNESS;
 }
 (*n)++;
 if (*n>MAXPTS)
   return(2);

/*  The latest coordinate was at the plate bottom, and it was stored
    in what now is x[*n-1] and z[*n-1].  Now we proceed to see if the
    next coordinate will be at the plate bottom, too (i.e., check if
    melting takes place at the next point along the bottom).
*/
if ( (temp=(*Model.temp_model)(x[(*n)-1]-stepsize,y,THICKNESS))>T_MELT )
{
  /*  Next point will also be at the plate bottom. */
  coord2.x = x[(*n)-1]-stepsize;
  coord2.y = y;
  coord2.z = THICKNESS;
}
else
{
  /* Next point will be the first one above the plate bottom (i.e.,
     within the plate) after one or more boundary points were at
     the bottom.
     The last plate-bottom point is coord1, and an initial guess
     for coord2 is at 45 degrees elevation from the plate bottom.
  */
  coord1.x = x[*n-1];
  coord1.y = y;
  coord1.z = z[*n-1];
  coord2.x = x[*n-1]-stepsize/sqrt(2.0);
  coord2.y = y;
  coord2.z = THICKNESS-stepsize/sqrt(2.0);
  coord2 = angsrch_melt(y_axis, coord1, coord2);
  if (!length(coord2))
```

```
            return(1);
        };
      };

#ifdef PRINTPOINTS
  printf("\n%3d %6.3lf %6.3lf", *n-1, x[*n-1], z[*n-1]);
#endif

  };
  /* End of while-loop; only one more point is to be recorded.
     By now, coord2 is located above the plate surface. The last boundary
     point to be recorded is determined as the intersection of the plate
     surface with a straight line segment connecting the last recorded
     point and coord2. This is the same approach as used previously in
     this function, where the weld pool reached the plate bottom.
  */
  x[*n] = x[*n-1]-x[*n-1]*(x[*n-1]-x[*n-2])/(z[*n-1]-z[*n-2]);
  z[*n] = 0.0;
  (*n)++;
  if (*n>MAXPTS)
    return(2);

  return(0);
#ifdef PRINTPOINTS
  printf("\n%3d %6.3lf %6.3lf", *n-1, x[*n-1], z[*n-1]);
#endif

};

int topsection(double z, double stepsize, double *x, double *y, int *n)
/* ---------------------------------------------------------------------
  Finds a top view cross section of the weld pool (i.e., a section in
  the plane normal to the y-axis).

  z:          Displacement of the cross section plane.
  stepsize:   Desired distance between the points along the pool boundary.
  *x, *y:     Pointers to two vectors, one containing the x-axis values
              and the other holding the y-axis values of the boundary
              coordinates, respectively. These vectors must be allocated
              before this function is called.
  *n:         Number of found boundary points, returned from the function.
  --------------------------------------------------------------------- */

{
  struct coordinates
    coord1 = {0.0, 0.0, 0.0},
    coord2 = {0.0, 0.0, 0.0},
    last_coord2;
  int i;
  double temp;
```

```
    /* Initial point found at the front of the pool, in the x-z plane.
       Global variable "Bead.radius" used in a first guess.
    */
    coord1.x = sqrt(Bead.radius*Bead.radius); /* NOTE: assumes z=0 */
    coord1.z = z;
    coord1 = linsrch_melt (x_axis, coord1);
    if (!length(coord1))
      return(1);

    x[0] = coord1.x;      /* First point stored as x and y coordinates. */
    y[0] = coord1.y;
    *n = 1;

  #ifdef PRINTPOINTS
    printf("\n\n=== Topsection points ===\n\n i:  x[i]:  y[i]:");
    printf("\n%3d %6.3lf %6.3lf", *n-1, x[*n-1], y[*n-1]);
  #endif

    Bead.radius = length(coord1); /* Update radius and dT/dr with new info. */
    Bead.dTdr = -V*I*ETA/(1000.0*2.0*PI*K*Bead.radius*Bead.radius);
    coord2 = coord1;
    coord2.y = stepsize;  /* Second point assumed to be to the "right"
                             of the first point (seen from moving torch). */
    coord2 = angsrch_melt(z_axis, coord1, coord2);
    if (!length(coord2))
      return(1);

    while (coord2.y>=0.0)  /* Find successive boundary points. */
    {
      x[*n] = coord2.x;
      y[*n] = coord2.y;
      (*n)++;
      if (*n>MAXPTS)
        return(2);
      last_coord2 = coord2;
      coord2 = extrapolate(coord1, coord2);
      coord1 = last_coord2;
      coord2 = angsrch_melt(z_axis, coord1, coord2);
      if (!length(coord2))
        return(1);

#ifdef PRINTPOINTS
  printf("\n%3d %6.3lf %6.3lf", *n-1, x[*n-1], y[*n-1]);
#endif

  };
  x[*n] = x[*n-1]+(x[*n-1]-coord2.x)*y[*n-1]/(coord2.y-y[*n-1]);
  y[*n] = 0.0;
  (*n)++;
  if (*n>MAXPTS)
    return(2);
```

```
#ifdef PRINTPOINTS
  printf("\n%3d %6.3lf %6.3lf", *n-1, x[*n-1], y[*n-1]);
#endif

  for (i=*n; i<2*(*n)-1; i++)
  {
    x[i] = x[2*(*n)-i-2];
    y[i] = -y[2*(*n)-i-2];

#ifdef PRINTPOINTS
  printf("\n%3d %6.3lf %6.3lf", i, x[i], y[i]);
#endif

  };
   *n = 2*(*n)-1;
   return(0);
};
```

FILE SRCHMELT.C

```
/*      -- File SRCHMELT.C

        Author: Kristinn Andersen
        Date:   August 8, 1990.

        Revisions:
                Kristinn Andersen, August 28, 1990.

        There are two primary functions in this file:

        linsrch_melt()  - searches along a line for melting boundary;
        angsrch_melt()  - searches along a secant for melting boundary.

        Furthermore, one auxiliary function is defined in this file:

        rotate_2d()     - rotates a given 2D-vector by a specified angle.
*/

#include <float.h>
#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include "wmodel.h"

#define TEMP_OVER_MELT(COORD) ((*Model.temp_model)(COORD.x,COORD.y,COORD.z)-Metal.T_melt)
#define SQR(X) (X*X)
#define LENGTH(COORD) (sqrt(COORD.x*COORD.x+COORD.y*COORD.y+COORD.z*COORD.z))
#define ADD_COORDS(C1,C2,CT) {CT.x=C1.x+C2.x; CT.y=C1.y+C2.y; CT.z=C1.z+C2.z;}

#define PI 3.1415927

enum cartesian {x_axis, y_axis, z_axis};

struct coordinates ORIGIN = {0.0, 0.0, 0.0};

struct coordinates linsrch_melt
  (enum cartesian axis, struct coordinates coord);
struct coordinates angsrch_melt
  (enum cartesian axis, struct coordinates coord1, struct coordinates coord2);

#define ITMAX 20
#define MAX_POOLDIM 10.0*Bead.radius
#define DTDR_STEP 2.0

void rotate_2d
      (double *u, double *v, double angle)
```

```
{
  double
    length = sqrt((*u)*(*u)+(*v)*(*v));

    *u = length * cos(angle);
    *v = length * sin(angle);
}

struct coordinates linsrch_melt
  (enum cartesian axis, struct coordinates coord)
{
  double dTdr_factor;           /* Used in init. est. of temp. gradient. */
  double *u, u1, u2, u_last,    /* Used to bracket bound.*/
            f1, f2, f_last;
  double brack_factor;          /* Successive bracket interval increases.*/
  int iteration;                /* Iteration index for boundary search.  */
  double tol=1.0e-4;            /* Max. error in boundary location       */
  double a,b,c,d,e,min1,min2;   /* Variables used in boundary search.    */
  double fa,fb,fc,p,q,r,s,tol1,xm;

  /* For a given axis to search along for melting,
     1) let u point to the axis variable, and
     2) determine how close to perpendicular the search axis is to the
        weld pool surface in the neighborhood of coord.
  */
  switch (axis)
  {
    case x_axis:
      u = &coord.x;
      dTdr_factor = fabs(coord.x/LENGTH(coord)) + 1.0e-3;
      break;
    case y_axis:
      u = &coord.y;
      dTdr_factor = fabs(coord.y/LENGTH(coord)) + 1.0e-3;
      break;
    case z_axis:
      u = &coord.z;
      dTdr_factor = fabs(coord.z/LENGTH(coord)) + 1.0e-3;
      break;
};

if (*u<0.0)
  dTdr_factor = -dTdr_factor;

/* --- Step 1:  Bracket the pool boundary, i.e., find two points on
                 the search axis, so that the pool boundary crosses the
                 axis between these points.
*/

/* First test if pool boundary is at the initial point, "coord" */
u1 = *u;
```

```
  f1 = TEMP_OVER_MELT(coord);
  u_last = u1;
  f_last = f1;

  /* Try a 2nd test point, either closer to or further from origin. */

  if (f1>0.0)
    u2 = 2.0*u1;
  else
    u2 = u1/2.0;
  *u = u2;
  f2 = TEMP_OVER_MELT(coord);

  /* Use previous 2 points to estimate successive step sizes */
  if (fabs(f1)<fabs(f2))
    brack_factor = 2.0*fabs(f1)*fabs((u2-u1)/(f2-f1));
  else
    brack_factor = 2.0*fabs(f2)*fabs((u2-u1)/(f2-f1));
 while ((fabs(*u)<MAX_POOLDIM)&&(f1*f2>0.0))
 {
   if (fabs(f1)<fabs(f2))
   {
     u_last = u1;
     f_last = f1;
     *u = u1 += brack_factor*(u1-u2);
     f1 = TEMP_OVER_MELT(coord);
 }
   else
   {
     u_last = u2;
     f_last = f2;
     *u = u2 += brack_factor*(u2-u1);
     f2 = TEMP_OVER_MELT(coord);
   };
};
 if (fabs(*u)>=MAX_POOLDIM)
   return(ORIGIN);

/* --- Step 2:  The pool boundary crosses the search axis somewhere
                between points u_last and *u.  Now, proceed and find this
                point.
*/

fa = f_last;
a = u_last;

if (f1*f_last<=0.0)
  fb = f1;
else
  fb = f2;
b = *u;
```

```
Bead.dTdr = -fabs((fb-fa)/(b-a));

fc=fb;
for (iteration=1;iteration<=ITMAX;iteration++)
{
  if (fb*fc > 0.0)
  {
    c=a; fc=fa; e=d=b-a;
  }
  if (fabs(fc) < fabs(fb))
  {
    a=b;   b=c;   c=a;
    fa=fb; fb=fc; fc=fa;
  }
  tol1=2.0*DBL_EPSILON*fabs(b)+0.5*tol;
  xm=0.5*(c-b);
  if (fabs(xm) <= tol1 || fb == 0.0)
  {
    *u = b;
    return(coord);
  };
  if (fabs(e) >= tol1 && fabs(fa) > fabs(fb))
  {
    s=fb/fa;
    if (a == c)
    {
      p=2.0*xm*s;
      q=1.0-s;
    }
    else
    {
      q=fa/fc; r=fb/fc;
      p=s*(2.0*xm*q*(q-r)-(b-a)*(r-1.0));
      q=(q-1.0)*(r-1.0)*(s-1.0);
    }
    if (p > 0.0)
      q = -q;
   p=fabs(p);
   min1=3.0*xm*q-fabs(tol1*q);
   min2=fabs(e*q);
   if (2.0*p < (min1 < min2 ? min1 : min2))
   {
     e=d; d=p/q;
   }
   else
   {
     d=xm; e=d;
   }
}
else
```

```
    {
      d=xm; e=d;
    }
    a=b;
    fa=fb;
    if (fabs(d) > tol1)
      b += d;
    else
      b += (xm > 0.0 ? fabs(tol1) : -fabs(tol1));
    *u = b;
    fb = TEMP_OVER_MELT(coord);
  };
  return(ORIGIN);
};
#undef ITMAX

#define ITMAX 20
#define MAX_POOLDIM 10.0*Bead.radius
#define DTDU_STEP 2.0   /* If larger -> faster, but more prone to crashing */

struct coordinates angsrch_melt
  (enum cartesian axis, struct coordinates coord1, struct coordinates coord2)
{
  double *u, *v, f_last, f1, f2;/* Used to bracket boundary */
  double brack_factor;          /* Successive bracket interval increases.*/
  int iteration;                /* Iteration index for boundary search.  */
  double tol;                   /* Max. error in boundary location       */
  double a,b,c,d,e,min1,min2;   /* Variables used in boundary search.    */
  double fa,fb,fc,p,q,r,s,tol1,xm;
  double theta, theta0, theta1, theta2, theta_last, dr;

  struct coordinates
    coord,      /* coordinate variable for misc. purposes */
    d_coord,    /* coord2-coord1, or another small addition to coord1 */
    coord_d;    /* coord1+d_coord */

  /* For a given plane to search in for melting,
     1) let u point to the axis variable, and
     2) determine how close to perpendicular the search axis is to the
        weld pool surface in the neighborhood of coord.
  */
  d_coord.x = coord2.x-coord1.x;
  d_coord.y = coord2.y-coord1.y;
  d_coord.z = coord2.z-coord1.z;

  dr = LENGTH(d_coord);
  tol = 1.0e-4*dr;

  switch (axis)
  {
    case x_axis:
```

```
      u = &d_coord.y;
      v = &d_coord.z;
      break;
    case y_axis:
      u = &d_coord.x;
      v = &d_coord.z;
      break;
    case z_axis:
      u = &d_coord.x;
      v = &d_coord.y;
      break;
  };

  /* --- Step 1:  Bracket the pool boundary, i.e., find two points on
                  the search axis, so that the pool boundary crosses the
                  axis between these points.
  */

  /* First test if pool boundary is at point coord1+d_coord */

  theta0 = atan2(*v, *u);
  theta1 = 0.0;
  coord_d = coord2;
  f1 = TEMP_OVER_MELT(coord_d);
  theta_last = theta1;
  f_last = f1;

  /* Using the estimate of the temperature gradient, the 2nd test point
     for pool boundary is calculated.  DTDU_STEP is typically >1.0, to
     (1) compensate for the fact that the gradient calculations are usually
     underestimates, and (2) attempt to find a point on the other side of
     the pool boundary in one step.

     Bead.dTdr is the temperature gradient at the weld boundary, orthogonal
     to the boundary surface, and it is always negative.  Initial estimation
     of theta2 is most efficient if the boundary tracing is directed so that
     dT/d_theta is positive.
  */
  theta2 = theta1+(f1/(dr*Bead.dTdr))*DTDU_STEP;
  rotate_2d(u,v,theta2+theta0);
  dr = LENGTH(d_coord);
  ADD_COORDS(coord1, d_coord, coord);
  f2 = TEMP_OVER_MELT(coord);

  /* Use previous 2 points to estimate successive step sizes. */
  brack_factor = DTDU_STEP*fabs(f1*(theta2-theta1)/(f2-f1));

  while (  (fabs(theta1)<PI-1.0e-3)
         &&(fabs(theta2)<PI-1.0e-3)
         &&(f1*f2>0.0) )
  {
```

```
      if (fabs(f1)<fabs(f2)) /* Bracketing on theta1-side is more promising. */
      {
        theta_last = theta1;
        f_last = f1;
        theta1 += brack_factor*(theta1-theta2);
        rotate_2d(u,v,theta1+theta0);
        ADD_COORDS(coord1, d_coord, coord);
        f1 = TEMP_OVER_MELT(coord);
      }
      else              /* Bracketing on the theta2-side is more promising */
      {
        theta_last = theta2;
        f_last = f2;
        theta2 = PI/2.0;          /* += brack_factor*(theta2-theta1); */
        rotate_2d(u,v,theta2+theta0);
        ADD_COORDS(coord1, d_coord, coord);
        f2 = TEMP_OVER_MELT(coord);
      };
    };
    if ( (fabs(theta1)>PI-1.0e-3) /* Quit if bracketing interval gets huge. */
      || (fabs(theta2)>PI-1.0e-3) )
      return(ORIGIN);


    /* --- Step 2:  The pool boundary crosses the search axis somewhere
                    between points u_last and *u.  Now, proceed and find this
                    point.
    */

    fa = f_last;
    a = theta_last;

    if (f1*f_last<=0.0)
    {
      fb = f1;
      b = theta1;
    }
    else
    {
      fb = f2;
      b = theta2;
    };

/*if ( (fabs(b)<PI/2.0) && (fabs(a)<PI/2.0) )
     Bead.dTdr = -fabs((fb-fa)/(dr*(b-a)));      Revise dT/dr */

   fc=fb;

   for (iteration=1;iteration<=ITMAX;iteration++)
   {
     if (fb*fc > 0.0)
```

```
    {
      c=a; fc=fa; e=d=b-a;
    }
    if (fabs(fc) < fabs(fb))
    {
      a=b;   b=c;   c=a;
      fa=fb; fb=fc; fc=fa;
    }
    tol1=2.0*DBL_EPSILON*fabs(b)+0.5*tol;
    xm=0.5*(c-b);
    if (fabs(xm) <= tol1 || fb == 0.0)
    {
      theta = b;
      return(coord);
    };
    if (fabs(e) >= tol1 && fabs(fa) > fabs(fb))
    {
      s=fb/fa;
      if (a == c)
      {
        p=2.0*xm*s;
        q=1.0-s;
      }
      else
      {
        q=fa/fc; r=fb/fc;
        p=s*(2.0*xm*q*(q-r)-(b-a)*(r-1.0));
        q=(q-1.0)*(r-1.0)*(s-1.0);
      }
      if (p > 0.0)
        q = -q;
      p=fabs(p);
      min1=3.0*xm*q-fabs(tol1*q);
      min2=fabs(e*q);
      if (2.0*p < (min1 < min2 ? min1 : min2))
      {
        e=d; d=p/q;
      }
      else
      {
        d=xm; e=d;
      }
    }
    else
    {
      d=xm; e=d;
    }
    a=b;
    fa=fb;
    if (fabs(d) > tol1)
      b += d;
```

```
    else
      b += (xm > 0.0 ? fabs(tol1) : -fabs(tol1));

    rotate_2d(u,v,b+theta0);
    ADD_COORDS(coord1, d_coord, coord);
    fb = TEMP_OVER_MELT(coord);
  };
  return(ORIGIN);
};
#undef ITMAX
```

FILE NUNES.C

```
/*      -- File NUNES.C

        Author: Kristinn Andersen
        Date:   August 27, 1990.

        Revisions:
                Kristinn Andersen, August 29, 1990.

        The function nunes_temp() returns the temperature at a given
        point coordinates, with respect to the moving arc.
        For this, the nunes_temp() uses the following functions:

        tsai_func()     - a function whose integrand yields tsai_temp;
        qsimp()         - simpson integration;
        trapzd()        - trapezoidal function, used by qsimp().

*/

#include <float.h>
#include <math.h>
#include <stdio.h>
#include "wmodel.h"

#define PI 3.1415927
#define EPSILON 1.0e-3
#define JMAX 20

#define X coord.x       /* Shorthand for coordinate notations */
#define Y coord.y
#define Z coord.z

struct coordinates coord;

double qsimp(double u1, double u2, struct coordinates coord);
double trapzd(double u1, double u2, int n, struct coordinates coord);
double tsai_func(double t, struct coordinates coord);
double nunes_src (double x, double y, double z);
double nunes_temp (double x, double y, double z);

double bessi0(double x);
double bessk0(double x);

double bessi0(double x)
{
        double  ax,ans;
        double  y;
```

```
        if ((ax=fabs(x)) < 3.75) {
                y=x/3.75;
                y*=y;
                ans=1.0+y*(3.5156229+y*(3.0899424+y*(1.2067492
                        +y*(0.2659732+y*(0.360768e-1+y*0.45813e-2)))));
        } else {
                y=3.75/ax;
                ans=(exp(ax)/sqrt(ax))*(0.39894228+y*(0.1328592e-1
                        +y*(0.225319e-2+y*(-0.157565e-2+y*(0.916281e-2
                        +y*(-0.2057706e-1+y*(0.2635537e-1+y*(-0.1647633e-1
                        +y*0.392377e-2)))))));
        }
        return ans;
}

double bessk0(double x)
{
        double y,ans;
        double bessi0(double x);

        if (x <= 2.0) {
                y=x*x/4.0;
                ans=(-log(x/2.0)*bessi0(x))+(-0.57721566+y*(0.42278420
                        +y*(0.23069756+y*(0.3488590e-1+y*(0.262698e-2
                        +y*(0.10750e-3+y*0.74e-5))))));
        } else {
                y=2.0/x;
                ans=(exp(-x)/sqrt(x))*(1.25331414+y*(-0.7832358e-1
                        +y*(0.2189568e-1+y*(-0.1062446e-1+y*(0.587872e-2
                        +y*(-0.251540e-2+y*0.53208e-3))))));
        }
        return ans;
}

double nunes_src (double x, double y, double z)
{
  double  x2 = x*x, y2 = y*y, z2 = z*z;
  double  r  = sqrt(x2+y2+z2);
  double  r2 = r*r;
  double  v2a = SPEED/(2.0*ALPHA);
  double  v2a2 = v2a*v2a;
  double  mult_factor, pole_factor;

  mult_factor = exp(-v2a*(r+x))/(2.0*PI*K*r) * 0.001;
  pole_factor = I*V*ETA*(1.0-Arc.line_contrib);

  if (Arc.dipole!=0.0)
    pole_factor += Arc.dipole  * ( (1.0+x/r)*v2a+x/r2);

  if (Arc.x_qpole!=0.0)
```

```
      pole_factor += Arc.x_qpole * ( (1.0+2.0*x/r+x2/r2)*v2a2
                                    +(-1.0+2.0*x/r+3.0*x2/r2)*v2a/r
                                    +(-1.0+3.0*x2/r2)/r2 );

     if (Arc.y_qpole!=0.0)
      pole_factor += Arc.y_qpole * ( v2a2*y2/r2
                                    +(-1.0+3.0*y2/r2)*v2a/r
                                     +(-1.0+3.0*y2/r2)/r2 );

    if (Arc.z_qpole!=0.0)
      pole_factor += Arc.z_qpole * ( v2a2*z2/r2
                                    +(-1.0+3.0*z2/r2)*v2a/r
                                    +(-1.0+3.0*z2/r2)/r2 );

    return (mult_factor*pole_factor);
  };

  #define MAX_PAIRS 100
  #define CONV_EPSILON 1.0e-4
  double nunes_temp (double x, double y, double z)
  {
    int i;
    double temp_rise, d_temp;

    temp_rise = nunes_src(x, y, z);

    for (i=1; i<MAX_PAIRS; i++)
    {
      d_temp = nunes_src(x, y ,z-(double)i*2.0*THICKNESS)
             + nunes_src(x, y, z+(double)i*2.0*THICKNESS);
      if (d_temp<CONV_EPSILON*temp_rise)
        return (T_INIT + temp_rise);
      else
        temp_rise += d_temp;
    };
  };
```

FILE TSAI.C

```
/*      -- File TSAI.C

        Author: Kristinn Andersen
        Date:   August 2, 1990.

        Revisions:
                Kristinn Andersen, August 5, 1990.

        The function tsai_temp() returns the temperature at a given
        point coordinates, with respect to the moving arc.
        For this, the tsai_temp() uses the following functions:

        tsai_func()     - a function whose integrand yields tsai_temp;
        qsimp()         - simpson integration;
        trapzd()        - trapezoidal function, used by qsimp().

        The temperature calculated by tsai_temp is obtained through
        integration of the function tsai_func.  In N. Tsai's Ph.D.
        dissertation (MIT) and equation (10) in his paper in the AWS
        Welding Journal (Dec. 1983, pp.346s-355s) a time function (of t") is
        integrated from 0 to t, where t is "sufficiently large" for the
        system to reach equilibrium (ideally t should approaches infinity,
        but for practical purposes t on the order of at least 30-60 seconds
        was sufficient).

        It turned out that the original time function was ill-behaved for
        standard numerical integration using equally spaced points.  This
        was because the function approached infinity rather steeply
        (approximately as 1/sqrt(t")) as t" approached zero.  To achieve
        desired accuracy in the integration of this function, the sampling
        density close to t"=0 has to be correspondingly high.  But because
        the integration algorithms used here are based on equal spacing
        throughout the entire integration interval, this results in an
        excessive number of sampling points and function evaluations.

        To cure this, the integral was transformed to a finite-interval
        transform.  Three different transforms were tried:

                1.  u = 1-sech(t"),      t" = sech-1(1-u),
                                         dt" = du / ((1-u)*sqrt(u*(2-u))),
                    - integrated from 0 to 1;

                2.  u = arctan(t"),      t" = tan(u),
                                         dt" = 1/cos2(u) du,
                    - integrated from 0 to pi/2;
```

```
            3.  u = sec-1(t"+1),     t" = sec(u)-1,
                                     dt" = (sin(u)/cos2(u)) du,
                - integrated from 0 to pi/2.

        As a "benchmark", evaluation of the temperature (i.e., call to
        tsai_temp() ) was carried out at 3 coordinate points in
        Inconel-718.  The computation times on an 8MHz XT machine,
        using EPSILON=1e-4 (defined in this file), were as follows:

                0.      Original equation:      T = 20 minutes (approx.),
                1.      u = 1-sech(t"):         T = 03:50 minutes,
                2.      u = arctan(t"):         T = 02:21 minutes,
                3.      u = sec-1(t"+1):        T = 00:04 minutes (4 secs).

        The fact that transformation (3) results in fastest computations
        was anticipated.  Notice that at t"=0, du/dt" is 0, 1, "infinity",
        for transformations 1 through 3, respectively.  Qualitatively,
        the steep slope of the original temperature function for t" close
        to zero is offset by the "infinite" slope of the 3rd transformation
        for the same range of t".  To be more precise, Taylor expansion
        of the transformed temperature equation at u=0 results in a finite
        value, rather than infinity!  The temperature equation boils down to

                [sin(u)/sqrt(sec(u)-1)] * finite_value

                = sqrt(2) * finite_value                at u = 0.

        Eliminating the need for integrating a function that approaches
        infinity is assumed to be the dominating reason for the significant
        computational speed improvement.

                                        - Kristinn Andersen.
*/

#include <float.h>
#include <math.h>
#include <stdio.h>
#include "wmodel.h"

#define PI 3.1415927
#define EPSILON 1.0e-3
#define JMAX 20

#define X coord.x       /* Shorthand for coordinate notations */
#define Y coord.y
#define Z coord.z

struct coordinates coord;

double qsimp(double u1, double u2, struct coordinates coord);
double trapzd(double u1, double u2, int n, struct coordinates coord);
```

```
double tsai_func(double t, struct coordinates coord);
double tsai_temp (double x, double y, double z);


double qsimp(double u1, double u2, struct coordinates coord)
/* ------------------------------------------------------
  Returns the integral of tsai_func() from u1 to u2.  This integral is
  used in tsai_temp() to calculate the temperature at the point (x,y,z)
  specified in the structure coord.

  u1, u2:       Lower and upper limits for integration of tsai_func().
                Should be 0 and PI/2, but to avoid division by zero, the
                integration range is truncated by a distance EPSILON
                at both limits.
  coord:        Structure containing (x,y,z) with respect to the moving
                electrode, where the metal temperature is to be calculated.

  ------------------------------------------------------ */
{
  int j;
  double s, st, ost, os;

  ost = os = -1.0e30;
  for (j=1; j<=JMAX; j++)
  {
    st = trapzd(u1, u2, j, coord);
    s = (4.0*st - ost)/3.0;
    if (fabs(s-os)<EPSILON*fabs(os))
      return s;
    os = s;
    ost = st;
  }
  printf ("Too many steps in routine QSIMP");
}

double trapzd(double u1, double u2, int n, struct coordinates coord)
/* ------------------------------------------------------------
  Trapezoidal integration routine used by qsimp().

  u1, u2:       Lower and upper limits for integration.
  n:            ...
  coord:        Structure containing (x,y,z) with respect to the moving
                electrode, where the metal temperature is to be calculated.
  ------------------------------------------------------------ */
{
  double t, tnm, sum, del;
  static double s;
  static long it;
  long j;

  if (n==1)
```

```
    {
      it = 1;
      return
        (s=0.5*(u2-u1)*( tsai_func(u1, coord)
                        + tsai_func(u2, coord) ));
    }
    else
    {
      tnm = (double)it;
      del = (u2-u1)/tnm;
      t = u1 + 0.5*del;
      for (sum=0.0, j=1; j<=it; j++, t+=del)
        sum += tsai_func(t, coord);
      it *= 2;
      s = 0.5*(s+(u2-u1)*sum/tnm);
      return(s);
    }
  }

  double tsai_func(double u, struct coordinates coord)
  /* ------------------------------------------------
    Function whose integral across the variable u yields the temperature
    at any specified point in the welded metal.

    u:     Integration variable.  The original model of Tsai used time, t",
           for this purpose (integrated from 0 to infinity).  Here,
           u=sec-1(t"+1), so the integral is transformed to the range from
           0 to 1.
    coord: Structure containing (x,y,z), the coordinates (wrt. the moving arc)
           at which the temperature is to be evaluated.
    ------------------------------------------------ */
 {
    double secu_1, multiplier, exponent, w, result;

    secu_1 = 1.0/cos(u) - 1.0;      /* sec(u)-1 */

    multiplier = sin(u)/( sqrt(secu_1)*cos(u)*cos(u)
                         *(2.0*ALPHA*secu_1+SIGMA*SIGMA) );
    exponent   = - ( (X+SPEED*secu_1)*(X+SPEED*secu_1) + Y*Y )
                   /(4.0*ALPHA*secu_1+2.0*SIGMA*SIGMA)
                 - Z*Z/(4.0*ALPHA*secu_1);
/* original equation:
        multiplier = 1.0/(sqrt(t)*(2.0*ALPHA*t+SIGMA*SIGMA));
        exponent   = - ((X+SPEED*t)*(X+SPEED*t)+Y*Y)
                   /(4.0*ALPHA*t+2.0*SIGMA*SIGMA)
                 - Z*Z/(4.0*ALPHA*t);
*/
    return (multiplier*exp(exponent));
}

double  tsai_temp (double x, double y, double z)
```

```
/* ------------------------------------------
  Returns the temperature (deg.F) at any specified point in the welded metal.
  All other functions in this file are used by this function.

  x,y,z:        Coordinates (inches) in the welded metal, moving with the
                welding electrode.
  ------------------------------------------ */
{
        double pwr_mult, temp_rise=0.0;
        double u_start=1.0e-6, u_end=PI/2.0-1.0e-6;

        coord.x = x;
        coord.y = y;
        coord.z = z;
        pwr_mult = V*I*ETA*ALPHA/(1000.0*PI*K*sqrt(4.0*PI*ALPHA));
        temp_rise = pwr_mult * qsimp(u_start, u_end, coord);
        return (T_INIT + temp_rise);
}
```

FILE WMODEL.H

```
/*      -- File WMODEL.H

        Author: Kristinn Andersen
        Date:   August 8, 1990.

        Revisions:
                Kristinn Andersen, November 15, 1990.

        Contains global declarations that are used in various files
        of the weld modeling program.
*/


#define PI 3.1415927

enum viewplane {xview, yview, zview};

extern int
  wmodel_version,
  metric;

extern struct videoconfig screen;

struct coordinates
{
  double x;
  double y;
  double z;
};

extern struct metal_attributes
{
  char   name[20];      /* Name of metal */
  double thickness;     /* Thickness of welded plate, mm */
  double T_init;        /* Initial plate temperature, ambient temp., C */
  double T_melt;        /* Melting temperature of metal, C */
  double thermal_cond;  /* Thermal conductivity of metal ("k"), [] */
  double thermal_diff;  /* Thermal diffusivity of metal ("alpha"), [] */
  double spec_heat;     /* Specific heat of metal ("c"), [] */
  double density;       /* Density of metal ("rho"), [] */
} Metal;
#define T_INIT Metal.T_init
#define T_MELT Metal.T_melt
#define K Metal.thermal_cond
#define ALPHA Metal.thermal_diff
#define THICKNESS Metal.thickness
```

```
extern struct arc_attributes
{
  double current;       /* Arc current, Amperes */
  double voltage;       /* Arc voltage, Volts */
  double efficiency;    /* Arc efficiency, 0-100% */
  double speed;         /* Arc travel speed, [] */
  double dispersion;    /* Standard deviation of gaussian outline, mm */
  double dipole;
  double x_qpole;
  double y_qpole;
  double z_qpole;
  double line_contrib;
} Arc;
#define I Arc.current
#define V Arc.voltage
#define ETA Arc.efficiency
#define SPEED Arc.speed
#define SIGMA Arc.dispersion

extern struct bead_attributes
{
  double radius;
  double dTdr;
  double topfront;
  double topside;
  double topback;
  double rootfront;
  double rootside;
  double rootback;
  double penetration;
} Bead;

extern struct cross_section
{
  enum viewplane view;
  double offset;
  double *ucoords;
  double *vcoords;
  int    ncoords;
  double umin;
  double umax;
  double vmin;
  double vmax;
} Section;

extern struct model_type
{
  char name[20];
  double (*temp_model)(double x, double y, double z);
} Model;
```

NASA

Report Documentation Page

1. Report No	2. Government Accession No.	3. Recipient's Catalog No.

4. Title and Subtitle

Intelligent GTAW Control, SBIR Phase II
Final Report

5. Report Date
Feb. 28, 1991

6. Performing Organization Code

7. Author(s)

Kristinn Andersen
Robert Joel Barnett
James F. Springfield
George E. Cook

8. Performing Organization Report No.

10. Work Unit No.

9. Performing Organization Name and Address

Mid-South Engineering, Inc.
2131 Belcourt Ave
Nashville, TN 37212

11. Contract or Grant No.
NAS8-37401

12. Sponsoring Agency Name and Address

National Aeronautics and Space Administration
Washington, DC 20546-0001

13. Type of Report and Period Covered
Final Report,
through Feb. 1991

14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

The prototype welding control computer, and associated software and hardware, are being prepared for delivery to NASA. The work during the past few weeks has been concentrated on final touches on both hardware and software. Demonstration of the system is currently being planned and prepared.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Gas Tungsten Arc Welding (GTAW), Weld Modeling	Unclassified - Unlimited

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of pages	22. Price
Unclassified	Unclassified	162	$ 0

NASA FORM 1626 OCT 86